Subject: AVAIL

Release: REV19.3

Date: November 4, 1983

1 New functionality

None.

2 Bugs fixed

AVAIL has been fixed so that only U (Use) access to the MFD is required when the AVAIL command is given with a partition name or "*". [POLERS #37777] Previously, LU (List and Use) access was required. Note that LU access is still required if a partition name is not supplied (i.e. the command is given with no arguments or a logical device number). This is because when no partition name is supplied AVAIL must do a search of the MFD, and that is what List access protects.

Of course, Use access must be available on CMDNC0 and SYSTEM, and R (Read) access must be available on the DSKRAT, CMDNC0>AVAIL.SAVE, and SYSTEM>DISCS.

In addition, AVAIL has been fixed to correctly report the number of records used on partitions which are not an exact multiple of 16 records in size. Previously the count could be off by as many as 15 records. This bug was most obvious on very full partitions since the available record count could be negative.

Subject: Common Backend

Release: 19.3

Date: 07/13/83

New Functionality:

The number of include files allowed in one compilation unit has been increased from 99 to 500 for the compilers using the 19.3 backend by modifying the processing of include file names.

The 19.3 backend has incorporated the allocators of SPL, PL1G, PL1 and PASCAL into a common allocator which is now part of the backend library. This has been done in order to avoid having multiple copies of similar code in shared memory as well as to make maintenance of the allocator easier. The f77 allocator will be merged into the common allocator at the next Rev.

char optimizations: Certain character assignment statements and concatenations have been recoded to use inline code sequences.

Object groups have been expanded from 62 to 252 words.

Problems Fixed:

All peters fixed in the 19.3 backend are documented in the associated front ends.
Other problems fixed :

1. Discrepancy between f77 and the other compilers dealing with whether the contents of the token_node.spelling field are quoted or not.
2. Builtin funtion TRIM(char_string,bit(2),char(1)) now works correctly if the third argument is unaligned.
3. Calls to the runtime allocation routine p$alc now use short or long integer arithmetic to compute request size as appropriate.

Outstanding Problems :

1. The expanded listing option does not work correctly after the first %PAGE in the source program.
2. For unaligned character string arrays , evaluation of addr(char_array(index - 1)) is not done correctly.
3. Incorrect evaluation of addresses of large bit string arrays caused by using short arithmetic for calculating bit offsets.
4. Certain forms of aliasing will cause improper optimization.
5. TRIM (char_string,bit(2),char(1)) still does not work if the char(1) is a parameter and is unaligned in the caller.

These problems will be fixed in a future revision.

Environment:

This common backend must be compiled with a Rev 19.3 SPL because of dependence on shared link frame code.

Because of the 19.3 SPL usage the use of 19.3 SFG is also required.

Installation and Build Procedures:

The common backend uses standard installation and build procedures. The common backend should be installed before any other compiler products are built because of their dependence on it.

Contents:

The BACKENDSRC directory is the source directory on the master disk for the TSI compiler common code generator and related tools.

The following source files are related to the common code generator:
ALLOCATOR.SPL
DELETE_BINARY_FILE.SPL
FAKE_PASCAL_TYPE.SPL
FAKE_SETRC1.SPL
FIND_PREVIOUS.PMA
FIND_TOKEN_VALUE.PMA
HASHIT.PMA
IMODE.TB
IMODE3.SPL
OCHAR.PMA
OPTIMIZER.SPL
PCODEI.SPL
PCODEV.SPL
RDLNSP.PMA
SIMPLIFY.SPL
SPARE_DYNTS.PMA
UPCASE.PMA
UTILITIES.SPL
VMODE.TB
VMODE3.SPL
FIND_NODE.PMA
WLIN.PMA
WRITE_CODE.SPL
WRITE_SYMTAB.SPL
WRITE_PP.SPL
ZFIL.PMA

The following insert files are related to the common code generator:
EMIT_INFO.INS.SPL
GET_TEMP_NODE.INS.SPL
GLOBAL.INS.SPL
IMODEOPS.INS.SPL
LOAD_LENGTH.INS.SPL

NOTICE.INS.SPL
OPCODES.INS.SPL
OPTIMIZE.INS.SPL
PASCAL_TOKENS.INS.SPL
PBWRITE.INS.SPL
PCODE_TABLE.INS.SPL
TEMP_DEFS.INS.SPL
UTILITYDCLS.INS.SPL

The following source files are related to TSI tools:
BUILDER.SPL
BUILDM.SPL
PL1GDF.SPL
DCL.INS.SPL

The following file is provided for use with the TSI tools:
ERRORTEXT

The following files are provided for building and installing the common backend:
BACKEND.BUILD.CPL
BACKEND.INSTALL.COMI

Usage:

INTCOM* will contain the insert files needed for the general master disk builds for TSI products.

The compilers using the common backend can use the files provided for the TSI tools in their master disk builds. The source files for both the common code generator and the TSI tools will not be necessary to master disk builds of the TSI products because these will be compiled during the master disk build of the common code generator.

INDEX>BACKEND is the directory which will contain the uninstalled common code generator libraries built for the master disk. It will also contain the tools built for the master disk. The libraries will be stored in the subdirectory LIB as BACKENDV.BIN and BACKENDI.BIN. The tools will be stored in the subdirectory TOOLS as BUILDER.SEG and BUILDM.SEG. The info file for the common backend .Backend.runi, will be stored in the directory INDEX>BACKEND>INFO.

LIB will contain the installed libraries for the common code generator. There will be two libraries--BACKENDV.BIN and BACKENDI.BIN--which represent the code generator in V-mode and I-mode respectively.

TOOLS will contain the installed tools of the common backend, BUILDER.SEG and BUILDM.SEG.

SUBJECT: BASICV

RELEASE: 19.3

DATE: 5/23/83

NEW FUNCTIONALITY

IF A USER IS EDITING A FILE, AND ATTEMPTS TO FILE IT ONLY TO DISCOVER A CONDITION SUCH AS A DISK FULL, THE USER WILL BE GIVEN THE OPTION OF DELETING UNNECESSARY FILES, AND RETURNING TO THE EDITING SESSION FILE INTACT. (59354)

USER VISIBLE BUG FIXES

SPAR 3000875 - MAT READ AND MAT WRITE DO NOT WORK CONSISTENTLY ON BINDA FILES. FIXED.

SPAR 3000086 - CHAIN FROM LARGE PROGRAM TO LARGE PROGRAM TAKES ILLEGAL SEGMENT NO. FIXED.

45744 - STATEMENT CHANGE B TO B$ WHEN B(0) IS EQUAL TO ZERO, CAUSES AN ENDLESS LOOP WHEN PRINT B$ IS EXECUTED. FIXED.

59083 - ENDLESS LOOP OCCURRED WHEN ADDING LARGE # OF RECORDS TO MIDAS FILE. FIXED.

59354 - WHILE EDITING SOURCE FILE AND TRYING TO FILE, IF UNABLE TO FILE DUE TO A CONDITION SUCH AS DISK FULL, FILE IS TRUNCATED TO NULL. FIXED.

INTERNAL BUG FIXES

None.

OUTSTANDING PROBLEMS

None.

ENVIRONMENT

Required to build Basicv

FTN

PMA
SEG

SUBJECT: CBL

RELEASE: 19.3

DATE: 09/05/83

CBL release 19.3 is an ANSI Cobol compiler validated at the low-intermediate level. Command-line syntax, including compiler options, can be displayed by typing "CBL -HELP". One of these options, "-OLD," is recommended only for users of Prime's current Cobol compiler (COBOL), who wish to retain certain features specific to that compiler (see "COMPRESSED/UNCOMPRESSED," and "runtime file assignments" in the CBL manual (DOC5039-183). CBL supports both MIDAS and MIDASPLUS.

DIFFERENCES BETWEEN CBL AND COBOL

Appendix L of the CBL manual contains information on differences between CBL and COBOL. Two important differences are:

1. For DBMS users, CBL requires new versions of the preprocessor and subschema compiler. The COBOL preprocessor, CDML, is replaced by the CBL preprocessor, CBLDML; the COBOL subschema compiler, CSUBS, is replaced by the CBL subschema compiler CBLSUBS.

2. Programs compiled with CBL will take the "at end" path in "read...next" statements only for the "at end" condition as required by the ANSI standard. Programs compiled with COBOL take the "at end" path for any condition causing file status to be non zero.

INSTALLATION SHARE PROCEDURE

The shared version of the CBL library, CBLLIB.BIN, is designed to work with either CBL or COBOL. Users of COBOL who intend on upgrading to CBL should replace their current share procedure for the COBOL library with the CBL library share procedure. An unshared version of the CBL library, NCBLLIB.BIN is also supplied.

FUNCTIONALITY

The following is a partial list of Prime CBL extensions to the ANSI standard and to COBOL.

- COMP-1, COMP-2 (floating point) data types
- Eight levels of subscripting are allowed

- o CORRESPONDING option for IF and COMPUTE
- o The ROUNDED and SIZE ERROR options allowed for numeric moves
- o Arithmetic expressions with .O TO DEPENDING, IF, MOVE, PERFORM, SET, in subscripting, and in arithmetic statements
- o CBL interfaces with the Prime Source Level Debugger (-DEBUG)
- o Underscore and Lower case are allowed in identifiers
- o Right margin (col 72) can be extended to column 160 (-RMARGIN)

OUTSTANDING PROBLEMS

#35390 Using a subscripted field as a subscript gererates a fatal error during compilation.

#41364 Test for low-values not working consistently.

#47500 "<>" used as a "not equal" operator is incorrectly treated as "<".

#54373 File status is not returned correctly when the file status

dataname is defined in the Linkage Section.

#56507 -DIAGSONLY option creates a binary file.

#60255 An empty error file is created when only OBSERVATIONS (severity 1 errors) are encountered, and the program is compiled with -SILENT option.

Exponentiation gives only COMP-2 precision.
Exponentiation fails for negative exponents.

DIVIDE and MULTIPLY CORRESPONDING do not work.

Switches (CBLSW0 thru CBLSW7) work only with the non-shared library.

PERFORM using an arithmetic expression for the BY value loops.

-RANGE option is not implemented.

Changes have been made to the build file to detect error and warnings during the build. Also the rev stamp has been changed.

Changes have been made to the build file to detect error and warnings during the build. Also the rev stamp has been changed.

Subject: COPY_DISK

Release: REV19.3

Date: November 4, 1983

1 Changes made to COPY_DISK at 19.3

2 New functionality

o Added additional question for 300MB fixed media.

Subject: DBG

Release: 19.3

Date: November 16, 1983

1 New Functionality

- Full debugging capability for programs compiled by the New COBOL (CBL) compiler.
- It is no longer necessary to fully qualify the source line number of an internal procedure or function for Pascal and PL1/G when one is debugging a different procedure from the same source file.

2 Problems Fixed

2.1 IARS and PCLERS Fixed

- PCLER#31621
  In the FTN evaluation environment, comparing two INTEGER*2 variables whose values are 18000 and -18000, respectively, will now produce the correct answer.
- SPAR#32922
  In the FTN evaluation environment, DBG would not accept the expression ".2". It did accept "0.2", however. This is now fixed.

2.2 Other Bugs Fixed

- Using the SOURCE command with the FIND or NFIND subcommand on a null character string causes an appropriate error message instead of an error in DBG.
- Attempting to print out the first two elements of a Pascal array by typing ": S[1.2]" (instead of ": S[1..2]") caused DBG to terminate with an Access Violation. This now causes an appropriate error message.
- When using the CALL command with COBOL alphabetic or

alphanumeric arguments. DBG would not pad the arguments with the required blanks. The padding is now performed.

- In some cases using value tracing, the value of a variable would change without DBG reporting it. This is now fixed.

- Giving a breakpoint specification with an empty action list causes the previous action list on that breakpoint (if it exists) to be deleted. This feature was not working but is now fixed.

2.3 Internal Modifications

- The name of the configuration file for the installed version of DBG has been changed from DBGSRC>INS>CONFIG_PROD.INS.PLP to DBGSRC>INS>CONFIG_SHARED.INS.PLP.

3 Outstanding Problems

- ALL ARITHMETIC exceptions detected by hardware will cause DBG to lose the user program's environment.

- DBG stack areas still remain unprotected from being overwritten by a user procedure which, due to its own errors, has developed bad addresses. Users should thoroughly investigate this possibility before assuming that DBG has a bug in it.

4 Environment

Revision 19.3 of DBG requires a revision 19.2 or later version of PRIMOS, a revision 19.2 or later version of SPL, and a revision 19.1 or later version of PASCAL.

5 Installation and Build Procedures

Standard: build using DBG.BUILD.CPL, install using DBG.INSTALL.COMI, and share using DBG.SHARE.CPL.

Note: DBG uses segments 4037 and 4036 and allocates temporary segments not occupied by the user program downward from 4037.

Subject : DBMS

Release : 19.3

Date : November 30, 1983

Rev 19.3 includes all enhancements/bug fixes for Rev 19.2 and lower. This is the first ROAM based DBMS release. The existing DBMS product has been converted to work with the underlying services and functions provided by the ROAM system. This integrated DBMS/ROAM provides the same set of services with greater convenience and enhanced performance. For more detailed information refer to the DBMS/ROAM ALPHA TEST DOCUMENTATION.

I. New functionality and Enhancements

o ROAM Impact

Performance

DBMS performance is improved due to the new ROAM concurrency control algorithm which greatly reduces transaction aborts in a multi-user update environment.

Convenience

Before-image recovery is system-wide and automatically initiated at system coldstart if necessary. After-image recovery allows the roll-forward of a single file or a group of files. The database administrator will have much more flexibility in the placement of schemas within the file system.

Migration of functionality

All the DBACP recovery and backup functionalities are now handled by ROAM.

DBMS SYSTEM FILES

A prominent user visible change in the new DBMS is that the "DBMS" ufds no longer exist. The concept of a master DBMS ufd is eliminated. The SCHDIR file is replaced by the ROAM maintained "recovery table", RCVTAB. The error files, DAERRS and DBMSE, the TRACE file and the SYSLOG files are now created in DBASLB. The function of DALIST is assumed by user profiles which identify the privileged group of users known as "database administrators".

Compilation

The schema compiler will no longer generate a segdir of the format "SDnnnn" in the special DBMS ufd. Instead, the schema compiler will generate the DBMS master segdir in the current ufd in the format x.dbms, where "x" is the schema name as specified in the DDL source file. The "-vol" option on the schema command line is now replaced by the option "-output treename", (abbrev "-o"). While the concept of a master ufd is gone, the concept of a master segdir is retained. The master segdir of a schema is the segdir initially created by the schema compiler (i.e. the schema table). Other segdirs created by the subschema compiler or DBACP are termed slave segdirs. When a ROAM file is created (e.g. by compiling a DBMS schema), it is assigned a unique two-word file id. This can be used in the same context as the old schema number, and will be recognized as a schema identifier by DBACP.

Default attributes

A newly created schema will aquire the following default ROAM attributes: concurrency (before-imaging or transaction rollback) is enabled; the save date is null (schema not yet saved by a ROAM utility); before-image recovery (force writes of before images) and after-imaging are disabled. DMLCP will refuse to invoke a schema for which after-imaging is enabled with a null save date. The user should allocate the files for a schema, perform some initial loading, turn on after-imaging, and then save the schema.

| o All DBMS software

| Support for up to 256 processes has been added.

| o DMLCP

| The schema table is now mapped to a system segment which is shared by all users of that schema. If the schema table is large and there are several users accessing it, the savings in working set size should be significant. A private segment is assigned to each user and is returned when that user exits DBMS; therefore, no shared memory space is required. The operating system returns the system segment when the last user of that schema exits DBMS.

| As part of FINDing a record for which set currency was <u>not</u> suppressed, set ownership for all sets in which the found record was a member was automatically updated. Since a disk access per set involved is required to obtain the owner info, and since the application does not always subsequently use that owner info, DMLCP now delays the updates (and therefore disk accesses) until the application actually requires them (if ever). The performance improvement from this delayed

disk access will vary depending on the application requirements.

An additional variable has been added to FTRACE output. A value of 2 or 6 for "SPRES FLAGS" means the owner info for the current set occurrence was in the delayed state at the time of the error.

c DBACP

NOTE: The reader is referred to the rev 19.3 M.R.U. document for a complete description of the following two new commands.

Examine Set

```
                [ SET set-name  ]              [ schema-name    ]
   EXAMINE      [               ] OF SCHEMA    [                ].
                [ SETS          ]              [ schema-number  ]
```

The Examine command provides the DBA with the facility to obtain relevant statistics concerning the current condition of a set (or all sets) in the schema. The data which is gathered includes: total number of owner records, total number of member records, the maximum, minimum and average number of members per owner, the distribution of members in set occurrences (e.g. how many have fewer than five members, how many have more than one thousand members), space utilization information regarding the total amount of used and free node space, the total number of leaf, index and free nodes, and some I/O estimates regarding the approximate number of accesses required to retrieve a member record.

Based on this information, Examine Set recommends a node size which might better utilize the space allocated and/or minimize disk accesses at run-time. It further allows the DBA to project the results of a Pack Set (see below), given different node sizes and loading factors, before actually running the command. (A "loading factor" is a measure of the average number of members packed on a set leaf node.)

Pack Set

```
                                    [ schema-name    ]
   PACK SET set_name OF SCHEMA      [                ].
                                    [ schema-number  ]
```

The Pack Set command allows the DBA to reorganize the specified set with (if desired) a new node size and/or loading factor, based on the particular requirements of the database. It will pack the nodes in the set to more efficiently use the allocated node space and will improve the

placement of set nodes on the disk.

The DBA may optionally perform an Examine Set at the beginning of Pack Set. The command then requests the DBA to specify a node size and a loading factor for each set list. DBACP proceeds to pack the set, based on the user's input. Leaf nodes are rewritten to be the new size and as full as the loading factor indicates.

EXPAND AREA(S) performance and functionality has been improved as follows:

Only area header information is displayed prior to the actual expand (no bucket is accessed).

The original area file is no longer copied if an incremental expand (same location and bucket size) is performed. An incremental expand merely appends the new buckets onto the end of the existing file; therefore, it eliminates the need for having disk space to accomodate two files until expand completion.

The "next free bucket" pointer now points to the newly allocated (empty) buckets.

Only the newly allocated area bucket headers are cleared.

On a non-incremental expand, the integrity of the buckets is checked and fatal errors are generated to report 1. an infinite loop in the area file, 2. an invalid free bucket pointer, 3. an invalid bucket header in bucket <number>, 4. an invalid record directory in bucket <number>, 5. area bucket is larger than the internal buffer (max 16K words). In addition, the command "WILL NOT EXPAND AREA area-name" is generated if the area file is locked by another user or there is not enough disk space for the expand.

A break handler has been added for EXPAND AREA(S). Please note, however, that interrupting an EXPAND is not recommended. In order to receive the full benefits of the expand, both the copy (or append if an incremental expand) and relinking phases should be allowed to complete.

If a break (control p) is received in the estimation phase, cleanup is done and the command is aborted.

If in copy phase (i.e. not an incremental expand), DBACP reports how many buckets have been copied and how many are left to be copied, and asks the user if (s)he wishes to continue. If the dba does not wish to continue, the command is aborted and the area file is exactly as it was prior to the expand command. If the

dba does wish to continue, the command proceeds.

If in the append phase (i.e. is an incremental expand), DBACP links the first new bucket, reports how many buckets have been linked (original number + 1) and how many buckets are left to be linked, and asks the dba if (s)he wishes to continue. If the dba does not wish to continue, the expand is terminated and only one extra bucket is available for DBMS use (thus terminating the command is not recommended). The allocated but unlinked space (the number of buckets "left to link") can subsequently be linked to the area file by executing another EXPAND.

If in the bucket linking phase, DBACP reports how many buckets have been linked and how many buckets are left to be linked, and asks the user if (s)he wishes to continue. If the user does not wish to continue, the command is terminated and only those buckets that have been linked are considered to be on the available space chain. Since unlinked buckets are viewed as "full", disk space will be wasted if the user opts to terminate the expand here. The buckets can be subsequently relinked by another EXPAND (of at least one block) or by a PACK of the area (resulting in at least one fragment being packed and therefore requiring bucket relinking).

NOTE: The incremental expand was measured to be 30 times faster than that of the pre-19.3 expand area on a 30 MB database; better performance will result when performing incremental expands on a larger database. If insufficient space is available in the current area file location, we recommend acquiring that space (via, say, a MOVE of that file to a volume with adequate space) prior to the EXPAND in order to take advantage of the incremental expand performance gains.

PACK AREA(S) performance and functionality has been improved as follows:

Only area header information is displayed prior to the actual pack (no bucket is accessed).

A single pass is done to remove bad DBKs, remove deleted DBKs, pack fragmented DBKs, remove inter-record gaps, and compute area statistics. Whereas pre-19.3 PACK AREA had a minimum of four area file passes, new PACK's minimum has been reduced to one pass.

The integrity of the buckets is checked while packing/re-linking and fatal errors are generated to report 1. invalid bucket header in bucket <number>, 2. invalid record directory in bucket <number>, 3. invalid

bucket number in bucket <number>, 4. fragmented record linked to itself in bucket <number>, and 5. area bucket is larger than the internal buffer (max 16K words). Any packing completed prior to the error is correct and retained, but subsequent packing is aborted once a fatal error is received.

After successfully completing the packing phase of PACK, DBACP displays the number of record occurrences deleted, number of record fragments deleted, and the number of record fragments packed. If any of those numbers is greater than zero, relinking of the buckets follows.

A break handler has been added (for PACK AREA(S)). Please note, however, that interruption of a PACK is not recommended. In order to receive the full benefits of the PACK, both the packing and relinking phases should be allowed to complete.

If a break (control-p) is received during the packing phase, DBACP reports the number of buckets packed and the number of buckets left to pack, and asks the dba if (s)he wishes to continue. If the dba does not wish to continue, the command is terminated and the number of record occurrences and fragments deleted as well as the number of fragments packed is displayed. If the dba does wish to continue, the command proceeds.

If the break is received during the linking phase, DBACP reports how many buckets have been linked and how many buckets are left to be linked, and asks the user if (s)he wishes to continue. If the user does not wish to continue, the command is terminated and only those buckets that have been linked are considered to be on the available space chain. Since unlinked buckets are viewed as "full", disk space will be wasted if the user opts to terminate the pack here. The buckets can be subsequently relinked by an EXPAND of the area (of at least one block).

Rev 19.3 PACK was measured to be five times faster than the 19.1 pack area on a 64MB area file. Typically the new PACK is twice as fast as Rev 19.2 pack area.

VERIFY AREA(S) now has a break handler to abort the command and return to DBACP "READY" prompt.

By virtue of the work done to EXPAND AREA described above, some performance improvement has been realized in EXPAND CALC and SET in that only those files which actually were expanded have the ROAF headers cleared.

o DUMP

At 19.3 DUMP.SEG has been modified to be an interactive dump facility, which dumps in decimal, octal and ascii format.

To invoke the facility type SEG DBMSLB>DUMP.SEG from any ufd. DUMP will query you for the pathname of the file to be dumped, print out the file attributes of the file, query for the desired segment number to be dumped and query for the desired command.

Valid commands are as follows:

```
A(CCESS) ADDR                          ADDRs are decimal integers
D(UMP)   START_ADDR  END_ADDR          Dumps to TTY
H(ELP)                                 Lists valid commands
N(EW)    ENTRY_NO                      Changes dump subfile number,
                                       octal (prefix 0), ascii ('')
O(UTPUT) FILE_NAME -OPTIONS            Dumps to specified file,
                                       valid options:
                                           -t turns TTY back on
                                           -a appends to file
P(RINT)  START_ADDR  END_ADDR          Dumps to TTY 21 lines at
                                       a time
Q(UIT)
```

Note that this new facility has a vertical dump format.

A sample session is as follows:

```
OK, seg dbmslb>dump.seg
ROAM.FILE TREENAME: pathed>schemas>cpres-db.dbms
File CPRES-DB.DBMS is an ACTIVE MASTER file.
After-image recovery: OFF
Before-image recovery: ON
Trans rollback: ON
Usage: DBMS
Last saved:
File_id: 9286
FILE ENTRY No: 1
CMD: o 1 8
#      0/       1:      -15408     :141720     'CP'
#      0/       2:      -11679     :151305     'PE'
#      0/       3:      -11347     :151655     'S-'
#      0/       4:      -15166     :142302     'DB'
#      0/       5:      -24416     :120240     '  '
#      0/       6:      -24416     :120240     '  '
#      0/       7:      -24416     :120240     '  '
#      0/       8:      -24416     :120240     '  '
CMD: q
```

II. Problems Fixed

DBACP

The following polers are obsolete due to the ROAM services : 37900, 46653, 81975, 33542, 34679, 25408, 48007, 46238

When allocating a very large database, DBACP will not return negative numbers for file sizes. [POLER #58656]

Expand area no longer results in excessive I/O (see description of improved expand in section I). [POLER #46238]

By virtue of the new PACK SET command, set node size is changeable; therefore results from expand set are predictable. [POLER #41637]

Verification of an area correctly prints no error message when the area is correct. [SPAR #3000669]

Verification of a schema now reports the true actual size of a database for any newly allocated schemas and for any existing schema against which DBACP EXPAND has been run. [POLER #40934]

DBUTL

Text of the DBUTL HELP option correctly states that the major codes are given for the MON command. [POLER #47696]

The DBUTL FIX command is now obsolete.

DMLCP

The DML MODIFY command will produce an error when an attempt is made to modify items not in the subschema, but which had been added to the schema via the schema editor. [POLER #70958]

DMLCP correctly handles an attempt to store, for the second time, a record with no duplicates allowed. [POLER #52834, SPAR #2004700]

FETCH NEXT DUPLICATE will execute correctly when the area was opened for retrieval. [POLER #44185]

DMLCP algorithm for reclaiming available space no longer overwrites valid data. Without this fix, valid data would be lost. If any 19.0 or 19.1 revision of DBMS has been installed on your system, PRIME support personnel will need to come on site to verify the integrity of the database. Please contact your support person, if such contact has not already been made. [POLER #60398]

DMLCP returns an error on an illegal store using a variable length record. [POLER #41982]

Transaction bit map will no longer overflow when there is a hung or very slow user. [POLER #29298]

CSUBS

CSUBS correctly handles records with no items. [SPAR #3000539]

CSUBS correctly processes a subschema with 30 character data names. [SPAR #3000144]

III. Problem Outstanding

DMLCP

Conversion between REAL*4 and packed decimal as data moves between schema and subschema is incorrect. [POLER #40392]

When deleting an owner record of a very large set, where the members are mandatory/automatic, an internal fatal error is encountered. [SPAR #3000714]

| FSUPS/CSUPS/CBLSUPS

| Write-access to a schema is required for compiling any of its
| subschemas. Write-access should be required when posting
| updates against a database.

IV. Environment

Rev 19.3 DBMS requires PRIMOS Rev 19.3, ROAM Rev 19.3 and SEC Rev 19.0 or greater. DBMS runtime (DMLCP) requires the use of shared segments 2001, 2002, 2003 and 2012 as well as private segments 4030, 4031, 4032, 4033 and 4034.

V. Installation and Build Procedures

Build -- standard Installation and general information to follow.

FILES ON SYSTEM TAPE

DBMSEX (ufd)

This is the primary packaged UFD supplied on the system tape. It is a prerequisite for all the others. By itself it is adequate to allow for execution only of DBMS. Its UFD structure is followed by each of the other packaged UFDs. Exceptions will be noted where appropriate. The only file at the top level is the install procedure DBMS.INSTALL.CPL. Unlike other products, DBMS has been granted an exception from the master disk standard to allow us to have the install files be CPL procedures in order to simplify the task. However, the install file, DBMS.INSTALL.COMI, of the whole product is now a COMI file.

DBMSEX>DBMSLB (sub-ufd)

Here we find the shareable run files, segment directories, and support utilities which will be moved to top level DBMSLB by DBMS.INSTALL.

DB2001 DB2002 DB2003 DB2012 DB4000
DB2070 VERYPRT.SAVE DUMP.SEG IDBMS.SEC
DBACP.SEG DBUTL.SEG SUMMARY.SAVE DAEERS
DBMSE

DBMSEX>COMNDS

This UFD holds the CPL interludes to those V-mode products found in DBMSLB which will become external commands to PRIMOS. These interludes are moved to the top level DBMSLB by DBMS.INSTALL.

DBACP.CPL DBUTL.CPL

DBMSEX>BINARY (sub-ufd)

Each sub-product in a package has a binary file in this UFD of the form %prod%.BIN. In addition we have the following files produced by LOAD_LIB:

DBDATA.BIN VERSIO.BIN LIBRARY.MAP

DBMSEX>JOBS (sub-ufd)

Here we have the procedures of the form %prod%_LOAD.CPL which produce the run files or segment directories in DBMSEX>DBMSLB for this package. In the case of DMLCP the routine is called LOAD_LIB.CPL and it in turn requires some additional files:

DBDATA.FTN VERSIO.FTN HTAB.INS.FTN DYNT.PMA

Also found here are two utilities for setting up special configuration options at the time of the install and automating reloads of subproducts. (See following sections INTRODUCTORY MESSAGE CONTROL and RELOADING SUBPRODUCTS.)

SETUP.CPL RESET_UFD.CPL

DBMSEX>INFO (sub-ufd)

This UFD contains the documentation for the current release of DBMS.

DBMSDEF (ufd)

This UFD has a similar structure to DBMSEX. It is packaged with DBMSEX for systems on which there will be a need to develop schemas, so the sub-products contained herein are SCHEMA and SCHDEC.

The install file DBMSDEF.INSTALL.CPL is here at the top level.

DBMSDEF>DBMSLB (sub-ufd)

SCHEMA.SEG SCHDEC.SEG

DBMSDEF>CMDNCO (sub-ufd)

SCHEMA.CPL SCHDEC.CPL

DBMSDEF>BINARY (sub-ufd)

SCHEMA.BIN SCHDEC.BIN

DBMSDEF>JOBS (sub-ufd)

SCHEMA_LOAD.CPL SCHDEC_LOAD.CPL

DBMSFTN (ufd)

This UFD is packaged with DBMSDEF and DBMSEX for those systems on which FTN applications will be developed. Thus we include the sub-products FSUBS and DML as well as the LIB entry point file DMLLIB. The only file at the top level is DBMSFTN.INSTALL.CPL.

DBMSFTN>DBMSLB (sub-ufd)

FSUBS.SEG DML.SEG

DBMSFTN>CMDNCO (sub-ufd)

FSUBS.CPL FDML.CPL

DBMSFTN>LIB (sub-ufd)

DMLLIB.BIN

DBMSFTN>BINARY (sub-ufd)

FSUBS.BIN DML.BIN

DBMSFTN>JOBS (sub-ufd)

Here in addition to FSUBS_LOAD.CPL and FDML_LOAD.CPL we have the two utilities FDML.CPL and FLOAD.CPL for use in precompiling and loading applications programs. (See following section CREATION OF A DML APPLICATION PROGRAM.) They are moved to the top level UFD DBMSLB by the INSTALL procedure.

DBMSCOB (ufd)

This UFD is packaged with DBMSDEF and DBMSEX for those systems on which COBOL applications will be developed. Thus we include the sub-products CSUBS and DML as well as the LIB entry point file DMLLIB. The only file at the top level is DBMSCOB.INSTALL.CPL.

DBMSCOB>DBMSLB (sub-ufd)

CSUBS.SEG DML.SEG

DBMSCOB>CMDNCO (sub-ufd)

CSUBS.CPL CDML.CPL

DBMSCOB>LIB (sub-ufd)

DMLLIB.BIN

DBMSCOB>BINARY (sub-ufd)

CSUBS.BIN DML.BIN

DBMSCOB>JOBS (sub-ufd)

Here in addition to CSUBS_LOAD.CPL and CDML_LOAD.CPL we have the two utilities CDML.CPL and CLOAD.CPL for use in precompiling and loading applications programs. (See following section CREATION OF A DML APPLICATION PROGRAM.) They are moved to the top level UFD DBMSLB by the INSTALL procedure.

DBMSCBL (ufd)

This UFD is packaged with DBMSDEF and DBMSEX for those systems on which the new CBL (ANSI 1974 COBOL) compiler is installed and will be used to develop applications. Thus we include the sub-products CBLSUBS and DML as well as the LIB entry point file DMLLIB. The only file at the top level is DBMSCBL.INSTALL.CPL.

DBMSCPL>DBMSLB (sub-ufd)

CBLSUBS.SEG DML.SEC

DBMSCBL>CMDNCO (sub-ufd)

CBLSUBS.CPL CBLDML.CPL

DBMSCBL>LIB (sub-ufd)

DMLLIB.BIN

DBMSCBL>BINARY (sub-ufd)

CBLSUBS.BIN DML.BIN

DBMSCBL>JOBS (sub-ufd)

Here in addition to CBLSUBS_LOAD.CPL and CBLDML_LOAD.CPL we have the two utilities CBLDML.CPL and CBLLOAD.CPL for use in precompiling and loading applications programs. (See following section CREATION OF A DML APPLICATION PROGRAM.) They are moved to the top level UFD DBMSLB by the INSTALL procedure.

DBMSLGCL (ufd)

This UFD is packaged with DBMSDEF and DBMSEX for use on systems where the schema editor is needed. The install file SCHED.INSTALL.CPL is here at the top level.

DBMSLGCL>DBMSLB (sub-ufd)

SCHED.SEG

DBMSLGCL>CMDNCO (sub-ufd)

SCHED.CPL

DBMSLGCL>BINARY (sub-ufd)

SCHED.BIN TEXTED.BIN

DBMSLGCL>JOBS (sub-ufd)

SCHED_LOAD.CPL

INSTRUCTIONS FOR INITIAL INSTALLATION OF DBMS

1. If you already have a version of DBMS on your system, see the section UPGRADING AN EXISTING DBMS INSTALLATION.

2. Restore the UFDs supplied on tape. These may be one or more of the following:

   DBMSFX DBMSDEF DBMSFTN DBMSCOB
   DBMSCBL DBMSLGCL

3. Once the various UFDs described above have been created, Attach to the UFD where you want DBMS to reside and:

   CREATE DBMSLB

4. Run SETUP if you want your configuration options to be non-standard. (See following section INTRODUCTORY MESSAGE CONTROL.)

5. If you did not run it already as part of step 5 above:

   CO DBMSFX>DBMS.INSTALL.COMI

6. Share DBMS from the system console thus:

   CO SYSTEM>DBMS.SHARE.COMI

   Be sure to put the above share command into C_PRMO in CMDNCO so that DBMS will be shared any time the system is cold-started. (See section DMLCP INSTALLATION)

7. Finally DELETE the UFD(s) restored from tape.

UPGRADING AN EXISTING DBMS INSTALLATION

1. Save all Scratas to tape.

2. If you are upgrading from REV 18.2 or later, you should get rid of the DBMSxxxxPIM packaged binary UFDs since they are no longer on the system tape and thus will remain with outdated binaries on your system until manually deleted.

3. Restore the UFDs supplied on tape. These may be one or more of the following:

   DBMSFX DBMSDFF DBMSFTN DBMSCOB
   DBMSCAL DBMSLGCL

4. Run SETUP if you want your configuration options to be non-standard. (See following section INTRODUCTORY MESSAGE CONTROL.)

5. If you did not run it already as part of step 5 above:

   CO DBMSFX>DBMS.INSTALL.COMI

6. As of Rev 19.0, the error message files DAERRS, DBMSE, are copied by the INSTALL procedure from DBMSFX>DBMSLB to the UFD DBMSLB.

7. Share DBMS from the system console thus:

   CO SYSTEM>DBMS.SHARE.COMI

   Be sure to put the above share command into C_DBMO in CMDNCO so that DBMS will be shared any time the system is cold-started. (See section DMLCP INSTALLATION)

8. Finally DELETE the UFD(s) restored from tape.

9. If this is the first time that a ROAM version of DBMS has been installed, complete the steps outlined in the section CONVERSION OF RAM DATABASE TO ROAM DATABASE.

DATA ADMINISTRATOR AUTHORIZATION

Every DBMS site must have the following User Profile Groups defined:

.ROAM_ADMIN	ROAM data administrators
.DBMS_ADMIN	DBMS data base administrators

Persons belonging to the group .DBMS_ADMIN are authorized as valid data administrators. Without such membership, a user may not use any of the DBACP commands which alter a database or display sensitive information (such as privacy keys). Persons who also belong to the .ROAM_ADMIN group are classified privileged administrators. These are data administrators who may bypass the various schema privacy locks when using DBACP. A privileged data administrator would be responsible for the management and integrity of the DBMS as a whole.

INTRODUCTORY MESSAGE CONTROL

It is possible to inhibit the printing of an introductory message at run-time when DBMS is INVOKEd. To suppress the introductory message, run the CPL procedure SETUP.CPL in DBMSFX>JOBS. This utility will ask you if you want the message and then make the appropriate change to VERSIO.FTN and reload DMLCP.

RELOADING PRODUCTS

There are times when a specific sub-product of DBMS needs to be reloaded, ie: the segment directory needs to be created anew. To do this it is possible to use the same job streams which were used in the original building of the components of DBMS. For this to work you must start with the packaged UFDs delivered on your system tape.

These job streams are to be found in the JOBS UFDs of the packages to which they belong and have the names >prod%_LOAD.CPL.

Note that if you have replaced any of the libraries (ULIB, CLIB, ILIB, TEXTED, DMLCP, ASI, or ASG), you should reload all the sub-products which use them in their loading procedures. If you would like to automate the remaining reloads, you can use RESET_UFD.CPL (see section on FILES on SYSTEM TAPE).

The load procedures for sub-products produce segment directories in the the UFD DBMSLB within their respective packages. To install a reloaded segdir you need only copy it up to the top level UFD DBMSLB. The shareable segments produced by LOAD_LIB.CPL should also be promoted to the top level DBMSLB but will not come into use until the next time the DBMS.SHARE.COMI command stream is run (see below).

DMLCP INSTALLATION

DMLCP requires the exclusive use of shared segments 2001 and 2012 and private segments 4030, 4031, 4032. To install the shared library version of the DML command processor, the following command must be executed from the system console:

```
CO SYSTEM>DBMS.SHARE.COMI
```

This command stream installs the DBMS shared library, shares and initializes the DBMS segments, and initializes the Ring 3 semaphores. This command should be incorporated into C_PRMO, the command file which is always run after a cold start.

CREATION OF A DML APPLICATION PROGRAM

Once a schema and subschema have been written and compiled and the data base files have been allocated with DBACP, you can write application programs for the data base in either COBOL or FORTRAN. The sequence used to transform the source code into executable code is as follows:

(1) Preprocess the source code with the host language preprocessor (FDML, CDML or CBLDML).

(2) Compile the output of the preprocessor (xxxxx.FTN, xxxxx.COBOL, or xxxxx.CBL) with the host language compiler.

(3) Link the binary output of the compiler to the DML command processor with the segmented loader SEG.

Command procedures to do these operations with either a COBOL, CBL or FTN program may be found in UFD DBMSLB called CDML.CPL, CLOAD.CPL, CBLDML.CPL, CBLLOAD.CPL, FDML.CPL, and FLOAD.CPL. (The installation utility selects these out of the JOE UFD of the appropriate package if supplied on the system tape and promotes them to the top level DBMSLB.)

For example; to compile and load a COBOL program called "PROG", execute the following command:

```
R DBMSLB>CDML PROG
```

In turn; to compile and load a CBL program called "PROG":

```
R DBMSLB>CBLDML PROG
```

Likewise, to compile and load a FORTRAN program called "PROG":

```
R DBMSLB>FDML PROG
```

The output files created when using CDML.CPL, CBLDML.CPL, or FDML.CPL on the source file "PROG" are:

```
PROG.LIST - The preprocessor and compiler listings.
PROG.BIN  - The binary file output by the compiler.
```

The output files from using CLOAD.CPL, CBLLOAD.CPL, or FLOAD.CPL with program "PROG" are:

```
PROG.MAP  - SEG program map.
PROG.SEG  - The segmented run file.
```

The resulting user program is executed with the command:

SEG PROG

ROAM Conversion

The conversion of existing RAM style schemas to ROAM format is straightforward. No check will be made for incomplete transactions before the schema is converted; if there is any doubt the user may perform a DBACP RECOVER SCHEMA operation before doing the conversion.

There is a backward conversion path to convert a ROAM file to the old RAM based system. The backward conversion path means that the user does not have to save the RAM based schema before converting to ROAM. A ROAM schema can be converted back to RAM at any point (in particular, an archived ROAM database can be converted to RAM). First, the forward conversion path will be specified.

1) The new ROAM and DBMS software is installed at the customer site.

2) The passwords are removed from all the PDBMS ufds (ROAM does not support passwords).

3) The names in the DALIST are added to the .DBMS_ADMIN and/or .ROAM_ADMIN groups as appropriate. User profiles are updated to reflect new group membership.

4) The new DBUTL command "ROAM" is used to convert each schema to a ROAM compatible format. All schemas should be converted at once; old style RAM schemas cannot use the new DBACP, SCHED, DMLCP, etc. The SDnnnn segdir name in the PDBMS ufd will be changed automatically to <schema-name>.DBMS (the names of all slave SDnnnn's will be changed too). Note that part of the conversion process will be the truncation of the AI and BI files of the schema. Once a schema has been converted to ROAM format, DBACP type AI files cannot be applied against it.

If after-imaging is enabled for the schema, then the converted schema has to be saved using one of the ROAM utilities before application programs can be run against it (the ROAM command will assign the newly converted schema a null save date).

One invisible aspect of the conversion process is the resetting of the ROAM shared system-wide transaction number in the ROAM before-image file header. The RAM transaction number in the old per-schema BI file is compared to the new ROAM system-wide trno; if it is greater than the ROAM trno is set to its value.

NOTE: No DML application programs need be recompiled to run against the ROAM based schema.

The backward conversion process is as follows:

1) Re-create the appropriate PDBMS ufds. Place error files, SCHDTR, etc. in the master PDBMS ufd. Do not password ufds yet.

2) An archived ROAM database must first be restored to active status before it can be converted back to RAM.

3) To start the conversion process, type "RAM schema_name" in the ROAM based DBUTL.

4) You will be prompted for the volumes to which the master and slave segdirs will reside under RAM. DBMS ufds (PDBMS) must exist on these volumes. You will have to copy the segdirs to these volumes yourself.

5) As part of the conversion process, the segdir names will be changed to SCNNNN. AI/BI files of minimum size will be allocated. Copy the segdirs to the appropriate DBMS volumes (as entered in point 4). Passwords can now be set for each DBMS ufd.

6) Install the old DBMS software.

7) Increase the size of the BI file (via DBACP).

8) Any application programs that were compiled under ROAM must be re-compiled to conform to the old RAM invoke call.

Subject: DELETE

Release: 19.3

Date: 11/4/83

1 New Functionality

The command has been extended to allow for deletion of all entries in a directory.

2 Problems Fixed at Rev19.3

Fixed delete so that when the user goes to delete a delete-protected directory and answers "no" to the verification question, then the directory will be deleted, and a message will say that the directory has been deleted.

3 Problems Fixed

Fixed printing of names in error messages. Added checking and error messages about unknown types of file system objects. Fixed bad reporting on errors.

4 Outstanding Problems

User gets lots of questions on same object. Multiple and extra error messages.

5 Environment

Needs PRIMOS 19.0.65 or greater. (19.0.respin)

6 Installation and Build Procedures

A copy of delete.run must be in cmdnc0 before starting master disk build. Needs fsul>b.build.cpl to be run first. Other than that it is standard.

7 Usage

The usage information for the command follows.

Usage: DELETE target_tree [options]

target_tree treename of object to be deleted.

options may be selected in any order from the list below.

Option descriptions:

-QUERY, -Q
Specifies that DELETE is to request that the user resolve unexpected or potentially dangerous situations. This is the default.

-NO_QUERY, -NQ
Specifies that DELETE is NOT to request the user's permission but to attempt to resolve those situations in the most intuitive fashion.

-FORCE
Specifies that DELETE is to force-delete a delete protect object. If "-force" is not specified, then DELETE will request the user's permission to force delete a protected object (unless "-no_query" was specified).

-REPORT, -RPT
Specifies that DELETE is report the results of each successful operation.

SUBJECT: DPTX PRODUCTS

RELEASE: 19.3

DATE: September 27, 1983

1 INTRODUCTION

DPTX at REV 19.3 is now only one product named DPTX. This product contains all functionality previously contained in DPTX-DSC, DPTX-TSF, and DPTX-TCF.

2 NEW FUNCTIONALITY

1. Support for the PST100 has been added. To use the PST100 terminal, the new command PST100DSC is used. PST100DSC is used in the same manner as PT45DSC.

2. Support for the PT46 terminal has been added. To use a PT46 terminal, the new command PT46DSC is used. The PT46 is an enhanced PT45 terminal and usage of PT46DSC is identical to the usage of PT45DSC.

3. DPTXMTR, the DPTX Status and Queue monitor has been added. This program displays information regarding DPTX queues or events on the communications lines. Usage of DPTXMTR is as follows:

   For Status Monitor: DPTXMTR [-TOTals] [-FREQuency <mins>]

   For Queue Monitor : DPTXMTR -QUEUE [-FREQuency <secs>]

4. A common error and status message module is now used by all DPTX DSC processes (DSCMAN, EM3270). This provides for enhanced error and status reporting, and finer granularity in timing of events occuring within the DPTX system.

5. DPTX startup files (i.e. BSC.COMI, DSCGO.SEG) are now located in the UFD DPTX* instead of UFD SYSTEM.

3 BUGS FIXED

3.1 USER VISIBLE BUG FIXES

1. A bug was fixed where TCF would not accept any PA key as the exit indicator. (POLERS 42967)

2. A bug was fixed where OWLDSC would send an SBA order with address, row 25, column 1, causing an 'Invalid Screen Address' error on BD$OUT. (POLERS 46862,47349,31863,29344)

3. OWLDSC will no longer leave the keyboard locked when it receives a transmission failed message from BD$INP. (POLERS 35639)

4. A Screen 'scramble' problem caused by EM3270 not checking for chained RR's on large data messages has been fixed. (POLERS 51126,51494)

5. PT45DSC will no longer change the DUPLEX settings when an invalid station name is given. (POLERS 51496)

6. A problem where the 3277 FORMS driver was incorrectly formatting the screen has been corrected. (POLERS 33095)

7. BD$LST would return incorrect information about the status of EM3270 control units. A device which was 'up' was reported 'down'. (POLERS 44586)

8. BD$LST is now a valid call, the DYNT in BDVLIB had been missing. (POLERS 51310)

9. Files spooled without the -FTN option will no longer print one line per page. (POLERS 41033)

10. A problem was fixed where BD$OUT was called to tranmit one character and VBTMPL.FTN would do character validation on the whole word (2 characters) causing an 'Illegal Character' error. (POLERS 43811)

11. A problem was fixed where OWLDSC would exit with 'Numeric Check' error when reading a numeric field which contained spaces. This fix also corrects a problem with TCF, where a HOST DOWN message appeared when reading a numeric field with host written non-numeric data. (POLERS 32428,60277)

12. The READ Modified command now works correctly if a Short Read AID is outstanding. (POLERS 32431)

13. The Duplex setting of a PT45DSC user will no longer be

changed if the user supplies an invalid station name. (POLERS 51496)

14. PT45DSC will no longer get a "ILLEGAL_SEG_NO" error when more than 500 fields are written. (POLERS 61948)

15. A problem where the DFTCFG did not process the VFC option correctly, which caused the option to be ignored in some instances, has been fixed. (POLERS 42971)

3.2 INTERNAL BUG FIXES

1. The Poll delay for TM3270 was being calculated incorectly, the delay should have been multiplied by 330, instead it was multiplied by 33. (POLERS 32418)

2. A change was made to DLDSO.FTN so that a new routine (LDTM01) is called instead of LOADQ1. TM3270 uses this routine so only 256 characters are inserted per data block with trailing SYNC characters. This is to simulate timefills.

3. A bug was fixed where BSCMAN retried a receive on a remote end timeout, now BSCMAN just informs the protocol handler. (POLERS 42972,42960)

4. A bug was fixed where BSCMAN would not reset the OUNLIN variable before logouts, causing code to be executed for lines used during previous invocations of BSCMAN.

5. The emulator will now send a complete read data stream when a short read AID is outstanding, and a read command is issued. A logical variable, CMDINP was added to inform the RDMODR sybroutine whether the call was generated by a poll or a command. (POLERS 32431)

6. A bug was fixed where an unlock was not being sent to the BDI when a NAK was received while the emulator was in response outstanding mode. (POLERS 35639)

7. New general poll logic has been added to work correcly in multi-dropped environment. This logic replaces MAXPOL and CUADDR. This causes DPTX to miss fewer polls. (POLERS 45649,31324,32424,42975,45647)

8. A bug was fixed where an EOT was output when DSR went down in Response Outstanding Mode. (POLERS 42913)

9. A problem was fixed where Data Blocks were being lost in Text Mode. (POLERS 42914)

10. A fix was made to the SCHCNT mechanism, SCHCNT is now

initialized to zero at EM3270 startup. (POLERS 42915)

11. A problem was corrected in the "not our device" logic. If an IR status was sent to a non-existing device, the subsequent ACK would reset someone else's Virtual Buffer. (POLERS 32426)

12. A bug was fixed in a TOOCT statement that caused a POINTER FAULT.

13. The call to SFUTN to release BSCMAN's timer in DPTOFF has been removed so BSCMAN can be notified until logout.

14. A change was made so that a NACK or an IR status only matters for DH devices. Otherwise the BDI would not work correctly.

15. Counters were added for Control Unit timeouts. When equal to three, the Control Unit is marked down. When a poll is output and the Control Unit does not respond, a NAK is sent.

16. A bug was fixed where TM3270 marked the Control Unit down incorrectly. (POLERS 32432)

17. EM3270 will no longer reset variables for devices it does not own. This problem would cause device status to be lost. (POLERS 32426)

18. The message output by EM3270 when there are no control units configured on the line has been corrected. Previously, EM3270 reported an error of 'Bad Message Key'. (POLERS 35998)

4 OUTSTANDING PROBLEMS

This section indicates, as of July 15, 1983, all outstanding problems which Engineering is aware of in REV 19.3 DPTX software. Problems which have been reported via the POLERS system have associated POLERS numbers with them.

1. As part of the DPTX/BSC product, PT45DSC transmits only data that has been modified. In certain cases this is inconsistent with the 3277 it emulates.

2. TM3270, communicating with COMMAND devices, as part of the DPTX/TSF and DPTX/TCF products, uses chained WRITES to acknowledge receipt of READ MODIFIED data. This is acceptable to the 3271 Mod 2. However, the second generation BSC control units (3274 and 3276) do not allow this. Because of this, it is

not possible to run DPTX/TSF with second generation control units.

3. PT45DSC, running as part of the DPTX/DSC product, will not run correctly at 9600 bps. Because the PT45 is a slow device compared to the DPTXF, the PT45's buffers can be overrun, resulting in 'broken' screen formats and scrambled messages (POLEDS 79480). The temporary solution to this is to run the PT45 at 4800 bps or to insure that the DMQ size for that line is set at its default value.

4. The ERASE INPUT key used on a terminal connected to an IBM control unit attached to the PRIME via DPTX/TSF will not cause the virtual buffer to be updated. This is only a problem for those using TCF. (POLEDS 75025)

5. Users of DPTX/TSF using DISLOG YES as a config option should be aware that TTYNCPing does not prevent the AMLDIM from performing logout abort checks when DISLOG YES is specified. The temporary work around is to not specify DISLOG YES as a config option. (POLEDS 42509)

6. In a WARM START condition, if a PRIME 7 user is attached to DPTX and is currently SELECTed, the 'hard lock' will not be reset. (An 'unlock' message is not sent to the SELECTed user.) The work around for this is to power the terminal off and back on again. The virtual buffer remains unchanged from the last successfully completed host transmision.

7. TM3270 can sometimes loose a DF status from a 3277 causing a 'one screen behind the current command' behavior. The way to clear this condition is to power off and power on the 3277.

8. The DPTX spooler does not recover from abnormal error conditions. If one of these conditions (i.e. paper out) occurs, the DPTX spooler will logout.

9. In some cases, some fields on an IBM XEDIT screen will not be transmitted by a PT45.

10. DPTX sometimes looses its internal buffering after running for long periods of time. The message 'Insufficient Buffering' may appear in any of the DPTX process output logs. The only way to clear this condition is to shut down DPTX and restart DPTX.

5 ENVIRONMENT

DPTX at REV 19.3 requires a REV 19.3 PRIMOS and SPOOLER.

6 INSTALLAZATION AND BUILD PROCEDURES

Standard.

EDB and LIBEDB for rev 19.3

DATE: June 29, 1983

TO: R & D Personnel

FROM: Kevin Cummings

SUBJECT: EDB and LIBEDB for rev 19.3

REFERENCE: None

KEYWORDS: EDB, LIBEDB, LIBRARIES

ABSTRACT

This document describes the change made to EDB and LIBEDB for PRIMOS release 19.3.

EDB and LIBEDB for rev 19.3

1 EDB

EDB now supports object text with blocksizes > 64 words per block. The maximum blocksize allowed by the object text (by design) is 256 word blocks.

Subject: EDIT_PROFILE

Release: REV19.3

Date: November 4, 1983

1 New functionality

o Added MINIMUM_PASSWORD_LENGTH command.

o Added feature to print name of SAD'S parent in LIST_SYSTEM command if not MFD.

o Added feature so that EDIT_PROFILE checks input for null group name: <none>.

o Improved HELP command

2 Internal Bug Fixes

o Improved error handling during initialization.

o Improved handling of damaged read/write locks.

Subject : RJE Phase 2 Emulator - EM1004

Release : 1.5

Date : 09/16/83

1. New Functionality

This is a replacement product. The RJOP is the new operator interface to all the emulators. The RJQ is the new utility for queuing files, and replaces the previous SEND utilities.

Other changes in functionality include :-

A) Destination detection has been added,
B) Dynamic SVLC line assignment has been added,
C) For receive translation, it is no longer necessary to tie up a terminal to run the emulator

For a description of new functionality, please refer to IDR4036.

2. Problems Fixed

None

3. Outstanding Problems

TAR/Poler 25127 : Untranslated Receive files, on delsite, will cause a receive file pointer error.

TAR/Poler 81776 : Time-outs occur when running at 1200 baud.

4. Environment

This release of the product requires PRIMOS 19.3.

5. Installation and Build Procedures

These are standard.

Subject : RJE Phase 2 Emulator - EM200UT

Release : 1.5

Date : 09/16/83

1. New Functionality

This is a replacement product. The RJOP is the new operator interface to all the emulators. The RJQ is the new utility for queuing files, and replaces the previous SEND utilities.

Other changes in functionality include :-

A) Destination detection has been added.
B) Dynamic SMLC line assignment has been added.
C) For receive translation, it is no longer necessary to tie up a terminal to run the emulator

For a description of new functionality, please refer to IDR4036.

2. Problems Fixed

None

3. Outstanding Problems

There are no known outstanding problems.

4. Environment

This release of the product requires PRIMOS 19.3.

5. Installation and Build Procedures

These are standard.

This release of the product requires PRIMOS 19.3.

5. Installation and Build Procedures

These are standard.

Subject : RJE Phase 2 Emulator - ENHASP

Release : 1.5

Date : 09/16/83

1. New Functionality

This is a replacement product. The RJOP is the new operator interface to all the emulators. The RJQ is the new utility for queuing files, and replaces the previous SEND utilities.

Other changes in functionality include :-

A) 8 lines are now supported.
B) Modem speeds of up to 56K2 are supported,
C) CPU usage has been reduced,
D) Higher line throughput has been achieved,
E) Destination detection has been added,
F) Debug facilities have been improved,
G) Dynamic SMLC line assignment has been added,
H) Multi-streaming is now supported,
I) Limited HOST mode is supported,
J) Improved line recovery has been added

For a description of new functionality, please refer to IDR4036.

2. Problems Fixed

None

3. Outstanding Problems

There are no known outstanding problems.

4. Environment

This release of the product requires PRIMOS 19.3.

5. Installation and Build Procedures

These are standard.

Subject : RJE Phase 2 Emulator - EMX80

Release : 1.5

Date : 09/16/83

1. New Functionality

This is a replacement product. The RJCP is the new operator interface to all the emulators. The RJC is the new utility for queuing files, and replaces the previous SEND utilities.

Other changes in functionality include :-

A) 8 lines are now supported.
B) Modem speeds of up to 56KB are supported.
C) CPU usage has been reduced.
D) Higher line throughput has been achieved.
E) Destination detection has been added.
F) Debug facilties have been improved.
G) Dynamic SMLC line assignment has been added.
H) Line recovery has been improved

For a description of new functionality, please refer to IDR4036.

2. Problems Fixed

Emulator Name	X80
Issue No.	1.5
TAR/POLER No.	---*---
Changed by	Suzy
Date Fixed	07/15/83
User Visibility	None

Fix

A "no progress" flag is set if the emulator gets into a situation where one end of the link is getting receive timeouts but is in the midst of bidding for the line.

3. Outstanding Problems

TAR/Poler 14206 : On receiving EOT, the emulator will abort transmission.

TAR/Poler 22404 : It is possible to lose data if the line is disabled in mid-file.

4. Environment

This release of the product requires PRIMOS 19.3.

5. Installation and Build Procedures

These are standard.

Subject : RJE Phase 2 Emulator - EMXPY

Release : 1.5

Date : 09/16/83

1. New Functionality

This is a new emulator.

For a description of new functionality, please refer to IDR4036.

2. Problems Fixed

Emulator Name	XPY
Issue No.	1.5
TAR/PCLER No.	---*---
Changed by	Suzy
Date Fixed	08/08/83
User Visibility	None

Fix

In the module XDN.PLP, the routine look_for_work has been modified so that, when devices go inoperable after sending a file, the device only becomes operable again when data is available for sending.

3. Outstanding Problems

There are no known outstanding problems.

4. Environment

This release of the product requires PRIMOS 19.3.

5. Installation and Build Procedures

These are standard.

Subject: F77

Release: 19.3

Date: November 29, 1983

1 New Functionality

1.1 New Warning Message

F77 will now generate a severity 1 error message when it encounters an attempt to compare floating point numbers for equality or inequality by, for example, use of the ".EQ." or ".NE." relational operators.

2 Problems Fixed -- User-reported

2.1 SOLERS 35864, 45126, 43025, 43203, 54458, 53043, 45146, 42710, 55678, 58228, 53395, 32959, 29079, 58283, 57407

Initialization of certain variables and arrays in DATA statements was performed incorrectly by the compiler. The entities involved included CHARACTER*1 arrays and odd-length CHARACTER variables. These problems have been fixed for revision 19.3.

2.2 SOLERS 41811, 47904, 47051

Error 444, "Statement ordering conflict", was issued at inappropriate times in certain function subprograms that had a length specification in their FUNCTION statements, e.g., "INTEGER*4 FUNCTION...". This problem has been fixed.

2.3 SOLERS 43211, 46662, 58974, 27231, 45793

The use of -XREF when compiling certain programs in -DEBUG mode produced an ACCESS_VIOLATION in DBG, when attempting to evaluate variables in COMMON blocks. This problem is now fixed.

2.4 POLERS 58114

An ACCESS_VIOLATION occurred in F77 when compiling a subroutine which used an octal PARAMETER to index into an array. This problem has been fixed.

2.5 POLERS 37812, 54751

List-directed output produced an incorrect number of blank lines in certain circumstances. It now works properly.

2.6 POLERS 34423, 58909, 35969

The compiler gave an error when it encountered a subscript of zero (0) in an array element name that was being initialized in a DATA statement, even when that name was legal according to the dimension of the array. This problem has been fixed.

2.7 POLERS 58913, 27229

F77 had a limit of only 99 total "$INSERT" files that could be included from one FORTRAN source file. This limit has been increased to 500 at revision 19.3.

2.8 POLERS 45825

F77 would not properly handle certain combinations of variable names, causing problems at load time. This situation has been corrected.

2.9 POLERS 27230, 5016

F77 has a limit in that it cannot properly report line numbers in a listing or via DBG for a FORTRAN source program that contains more than a total of 32,767 lines (lines in any "$INSERT" files are included in this total). At revision 19.3, the compiler issues a warning attesting to this fact if it encounters more than this number of source lines.

2.10 POLERS 31163

Error 442, "An array is used illegally as an operand", was issued inappropriately for a certain erroneous construct. A correct error message is now issued.

2.11 POLERS 40732, 54350

The storage class of those symbolic names that were related together by an EQUIVALENCE statement was not set correctly in certain situations where one of them was given a storage class of static. The compiler will now propagate the proper storage class throughout all symbolic names related together via EQUIVALENCE. The cross-reference listing format has been modified to correctly reflect this information. Another problem involving array storage and EQUIVALENCE statements was reported in POLERS 54350, and has also been fixed.

2.12 POLERS 41641

A parenthesis appearing in a PROGRAM statement caused the compiler to abort. Illegal syntax in PROGRAM statements is now reported in a severity 2 error message.

2.13 POLERS 48036

A FORTRAN statement that assigned a non-LOGICAL constant to a symbolic name of type LOGICAL was not flagged as an error by F77. It now is.

2.14 POLERS 57230

This POLERS reported a compiler abort when a CHARACTER*(*) variable appeared in an I/O implied-DO statement. This actual problem could not be reproduced within engineering, but a related problem was found and corrected.

2.15 POLERS 56490, 58265, 45430

Error messages were given by F77 stating that the user should correct all previous errors and re-compile, when in fact there had been no previous errors. The cause for this situation was identical for all three of these POLERS, and has been fixed.

2.16 POLERS 59137

F77 generated an erroneous error message for a substring name, where the substring name referenced a CHARACTER*(*) function name from within that function. This erroneous message is no longer generated.

2.17 POLERS 45632

Octal constants used in PARAMETER statements were not evaluated correctly by the compiler at run time. They now are.

2.18 POLERS 57175

This POLERS reported a problem concerning EQUIVALENCE statements that used to work at a previous revision of F77. This problem is now fixed correctly.

2.19 POLERS 42727

F77 was unable to initialize certain data with octal numbers. It now can.

2.20 POLERS 31192

A FORTRAN statement of the form 'READ(UNIT,FORMAT,100)' threw the compiler into an endless diagnostic loop. It now correctly identifies and reports such syntax problems without cascading error messages.

2.21 POLERS 47235

Internal READ or WRITE statements that don't reference a FORMAT caused F77 to emit a call to a non-existent library routine. The compiler has been fixed to generate an appropriate severity 3 error message for this situation.

3 Problems Fixed -- Internal

3.1 Substring names and implied DO-loops

The compiler issued an erroneous error message for a legal CHARACTER substring name that appeared in an implied DO-loop, when the name's substring expressions were controlled by the implied DO-loop variable. The message is no longer issued in this case.

3.2 F77 and DBG

The compiler did not emit information to the debugger about the first statement of an F77 main program if that statement were executable. The symptom of this problem was that no breakpoints could be set on such statements. This problem has been fixed.

3.3 CALL statement

F77 aborted with an ACCESS_VIOLATION% when it encountered a CALL statement with no name to call. An appropriate error message is now given.

3.4 CHARACTER*(*) FUNCTION

F77 did not generate proper code to access the length of CHARACTER*(*) functions from within them when an argument list was present. It now does.

3.5 ERROR 369

The severity of this error message, which states that "An argument to the <x> intrinsic has an inappropriate data type", was reduced from 4 to 3. The text of the message has been slightly modified also.

3.6 F77DATA file left open

A problem was fixed where the compiler did not close all its files under certain error conditions.

4 Already fixed:

These PCLRS were determined to be already fixed at the time they were received by the F77 engineering staff.

35196 - fixed at Rev. 19.1
35199 - fixed at Rev. 19.2
35905 - fixed at Rev. 19.2
35764 - fixed at Rev. 18.4
44000 - fixed at Rev. 19.2
45145 - fixed at Rev. 19.2
45574 - fixed at Rev. 19.1
45592 - fixed at Rev. 19.2
45635 - fixed at Rev. 19.2
47054 - fixed at Rev. 18.3
47574 - fixed at Rev. 19.2
52152 - fixed at Rev. 19.2
58265 - fixed at Rev. 19.2
58992 - fixed at Rev. 19.2
59091 - fixed at Rev. 19.2
60696 - fixed at Rev. 19.0

5 Referred

The following SOLERS were found to be problems with other software systems and were therefore sent to the appropriate engineering staff. Please consult the appropriate Release Documents as to their disposition.

Referred to the Libraries support staff:

29078
41894 - TU and TL can give bad results if next T is in a backward direction from previous T.
41976 - STATUS='DELETE' doesn't work if current UFD is passworded
42884
43541 - Internal writes are a problem with optimization.
44162 - F4IO B-format parser problem.
44606
47113 - Backspace on externally opened files doesn't work.
47265 - Problem with returned error codes.
47485
51145 - ERR= problem in I/O statement.
53111 - G-format problem.
55672
56601 - DATA statement format problem.
58118
58305 - F4IO problem concerning list directed array reads.
59085
60233 - BLANK='ZERO' option not working properly in OPEN statement.
60559 - Tape I/O problem.

The following SOLERS were referred to the common Backend engineering support staff:

30047 - Problem with IMODE.
31199 - Problem with IMODE.
32841 - Difficulty in firing value node for an expression which is subscript to a character array.
40695 - Bug when accessing arrays with mixed mode expressions.
43546 - "
47139 - "
45639 - IMODE array dimension problem.
47754 - Command line option -EXFLIST signals error condition.
55453 - A .DMSEPT file is taken by DPG to be program name.

The following SOLERS were referred to the Technical Publications department for documentation correction:

29073 - Outline correct size units on RECL parameter in OPEN statement.
29080 - State that FORMAT statement doesn't allow character expressions past 255 characters in length.
44313 - State that arrays in COMMON greater than one segment must be redefined to the same dimensions as they were

originally defined.

57119
58122 - Better documentation concerning location buffers.

6 Outstanding Problems

6.1 PDLRS 46975, 37173

DBG does not currently recognize F77 alternate entry points.

6.2 PDLRS 47463

F77 does not currently generate code that detects integer zero divide at run time.

6.3 PDLRS 44321

A legal DIMENSION statement causes the compiler to give an erroneous error message when the DIMENSION statement follows an ENTRY statement which immediately follows a SUBROUTINE statement.

7 Environment

The F77 compiler for revision 19.3 must be run on a revision 19.3 PRIMOS operating system installed on a 50-series and up computer.

8 Installation and Build Procedures

The F77 compiler requires the 19.3 master disk to build. The BACKEND, PMA, and SPL products must be built first.

Subject : RJE Phase 2 Emulator - F\GRTS

Release : 1.5

Date : 09/16/83

1. New Functionality

This is a replacement product. The RJOP is the new operator interface to all the emulators. The RJQ is the new utility for queuing files, and replaces the previous SEND utilities.

Other changes in functionality include :-

A) 8 lines are now supported.
B) Modem speeds of up to 56KB are supported.
C) CPU usage has been reduced.
D) Higher line throughput has been achieved.
E) Destination detection has been added.
F) Debug facilties have been improved.
G) Dynamic SMLC line assignment has been added.
H) Line recovery has been improved

For a description of new functionality, please refer to IDR4036.

2. Problems Fixed

Initial Release.

Emulator Name	GRTS
Issue No.	1.5
TAR/POLFR No.	-----
Changed by	Suzy
Date Fixed	09/13/83
User Visibility	None.
Fix	A change has been made to the module RX_RLOCK.PLP to ensure that long line of printer data do not loose the last character on a line.

3. Outstanding Problems

There are no known outstanding problems.

4. Environment

Subject: FED

Release: 2.1

Date: 28th November 1983

Abstract

This document contains information on release 2.1 of FED (the Forms Editor).

FED is supported in IRD&E Ireland.

The contact within marketing for FED is Ian Schmidt.

New functionality

This release sees some significant changes to FED.

1 Support for new FORMS features

At PRIMOS revision 19.3, the FORMS package has been enhanced to provide a multiple FORMS catalogue facility. This release of FED includes minimal support for the multiple catalogue facility using the following method.

A new option '-CATalog' (the short form is in upper case), has been added to the command line. The option is followed by the treename of the FORMS catalogue you wish to use with any 'CATALOG' operations you perform during this FED session. It is not possible to respecify the catalogue from within FED. If this option is omitted, the default catalogue FORMS* will be used.

The revised format of the command line is:

FED [-PROfile <profile_tree>] [-CATalog <catalog_tree>]

If <profile_tree> does not exist, FED will create it. If <catalog_tree> does not exist FED will return to PRIMOS with an error.

2 Modify

Some changes have been made to the MODIFY option. The changes are described below.

2.1 Cursor positioning

It has been noted that in many cases it would be useful to leave the cursor over the last field that was modified, rather than return it to the 'Command: ' prompt. Where appropriate, the cursor is now left over the last field modified.

2.2 'COMMAND' function key

As the cursor is no longer returned to the 'Command: ' prompt in all cases, a new function key has been added to place the cursor in the command field ready for the the next command to be entered. This function key will be allocated an unused number and added to existing profiles automatically by FED.

3 User friendly quitting

With previous releases of FED, pressing either of the 'return' function keys caused an implicit quit, and any changes made since the last save would be lost.

FED has been updated to note when a quit would cause loss of updates, and now offers the choice of quitting and losing the updates, saving the updates and then quitting, or returning to the function and not quitting at all.

4 Changes to field detailing

Some changes have been made to the detail fields option available from both CREATE and MODIFY. The two-line working window has been reduced to a single work line, and any redefinition of fields is achieved by overwriting the copy of the chosen line in the workline. If fields are added to the line, FED now makes an 'intelligent' guess as to which was the original field and which are the new fields. Finally, the scope for field type respecification has been increased.

Problems Fixed

None.

Outstanding Problems

None.

Outstanding Wishes

None.

Environment

FED 2.1 requires FORMS 19.3 for the latest versions of run-time, FDL, and FAP. FORMS 19.3 must be built before FED 2.1 will build correctly, and FORMS 19.3 must be installed and shared before FED 2.1 will run correctly.

Build and Installation Procedures

Standard.

For the installation of FED, three files are provided:

FED.INSTALL.COMI
FED.INITINSTALL.COMI
FED.INSTALL.CPL

If FED already exists on the system, the comi file FED.INSTALL.COMI should be executed to install the product. If this is the first time FED is to be installed on this system, the comi file FED.INITINSTALL.COMI should be executed to install the product. The initinstall file has the extra task of creating the sytem level ufd FED* over and above the tasks performed by the normal install file.

Subject: FIX_DISK

Release: Rev 19.3

Date: November 29,1983

0.1 Changes to FIX_DISK

The utility FIX_DISK has been changed to have the following new arguments:

1. ~-ADD_BADSPOT~<record_number>~;~-ADPADS~<record_number>~: Manual addition of a badspot. This option will allow users to specify new badspots. FIX_DISK will then add these badspots to the badspot file; and will attempt to remap the bad record to a new location.

2. ~-TRUNCATE~;~-TRU~: Truncate the file after an uncorrectable badspot. Normally FIX_DISK will create a null record in a good portion of the disk (record filled with zeros) for that record that can not be read, and will fix the header pointers such that the remaining records in that file are appended to the null record. When this option is specified, FIX_DISK will truncate the remaining records of a file when it finds a record that it can not correctly read. This is what FIX_DISK currently does.

3. ~-NUMBER_OF_RETRIES~<number>~;~-NUMRTY~<number>~: The number of times FIX_DISK will attempt to read a failing disk record. Each attempt includes all nine different ways of reading a record. (For details on the nine ways see the section on changes to PRIMOS.) The default will be ten retries.

FIX_DISK has also been changed to remap records on any errors. When FIX_DISK detects a soft error (one that can be read within the specified retrys), FIX_DISK will then copy that record to a new location and then relink all of the appropriate file header pointers.

FIX_DISK has been changed to do multiple read attempts on a failing record. This increases the likelihood that the record can be read, but does not guarantee that the record will be read.

FIX_DISK has been changed to allow badspots that are not part of the logical disk. This allows the badspot file that is in each logical disk of a physical disk to contain the same information.

The new version of FIX_DISK will only fully run on a revision 19.3 PRIMOS. It will continue to work under previous versions of PRIMOS with reduced functionality.

0.2 Installation

The installation FIX_DISK will follow standard procedures.

Product: FORMS

Release: 19.3

Date: 28th November 1983

Abstract

This document contains the information on release 19.3 of the FORMS system.

1 Introduction to changes and new features

Changes have been made to the FORMS package to improve performance, to improve the functionality and to improve the documentation. The changes are introduced in this section. They are described in detail in subsequent sections.

1.1 New catalog structure

The existing method of storing and accessing a form uses a hashing algorithm to convert from the formname to a 4 digit number. This number is then used as part of the filename for the form which is maintained in the sub-ufd LNK.LD in FORMS*. In a system using a large number of forms, a significant number of collisions can occur, thus increasing considerably the time required to access the form.

The hashing algorithm has been replaced by a storage method using the complete formname and device name.

The format of the new catalog is that a sub-ufd is maintained in FORMS* for each terminal type (with names, for example, of PST100 and PT45). The actual formname is now used within each of these sub-ufd's as the filename for the form.

There is no effect on the runtime directives for the user. The forms in the existing catalog are copied at system installation time to the new sub-ufd's in FORMS*. The existing catalog is not disturbed.

1.2 Directive terminator

The existing runtime routines read from the beginning of a runtime directive to the end of the buffer. In COBOL, where large buffers may be specified, this can be extremely time-consuming. (The problem does not arise in FORTRAN programs).

The runtime routines have been changed to accept runtime directives either with or without a terminating character. The terminating character chosen is an exclamation mark (!), the ASCII character code of which is 241(octal).

There is no effect on programs using the existing runtime directives. The addition of an exclamation mark to indicate the termination of a directive will result in faster execution of a COBOL program.

1.3 Multiple catalogs

User-defined FORMS catalogs are supported.

A user may create a FORMS catalog in one or more ufds on the system. Using FAP he may then add forms into this catalog, and may then reference forms in this catalog during runtime.

A new runtime directive, ##catalog, is supplied to allow the application programmer to specify which catalog the form should be invoked from.

A new FAP command, catalog, is supplied to indicate which catalog the FAP commands work on.

1.4 Spool options

The ##print runtime directive supports the following 'spool' options
AT - allows printing at the specified printer
COPIES - specifies the number of times the file is to printed
FORM - denotes the type of paper required
NOHEAD - omits the header and trailer pages

The current print directive causes a file to be placed on the spool queue each time it is called. This is highly inefficient if many screenfuls are to be printed. Three new runtime directives are provided to write screen data to a file and spool it when completed.

These are:
##log-open
##log
##log-close

1.5 Pre Fixes

All Polers outstanding at PRIMOS revision 19.1 are cleared with this release. These include removing the use of fixed file unit numbers from the runtime code, which currently causes interaction problems with MIDAS, and removing the setting and resetting of the XON/XOFF bits by the terminal device drivers.

1.6 Compatibility issues

The internal structure of the FORMS catalog is changed, and a conversion utility may be executed during FORMS installation to copy the existing forms to the new catalog. The conversion utility does NOT delete the existing FORMS catalog in LNK.FD.

Existing applications using shared FORMS may then run unchanged

using the updated catalog. Applications using non-shared FORMS must of course be relocated to include the updated FORMS, and the load sequence must also include the application library. (See the examples in section 3).

Migration to the updated FORMS may proceed on a piecemeal basis if required by using one FORMS system as the shared FORMS, and the other FORMS system as the non-shared FORMS (see section 3).

This updated FORMS works with PRIMOS revision 19.1 onwards.

1.7 Documentation

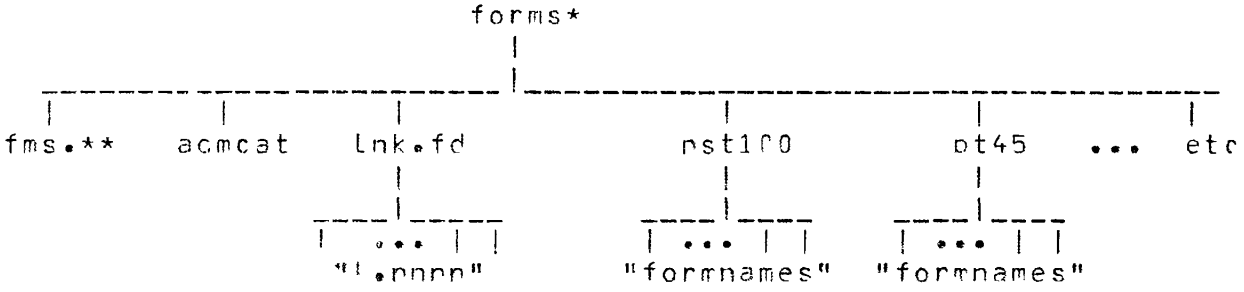
A new Prime Technical Update (PTU) is published, PTU2600 110, describing the upoated features. This also replaces the 3 existing PTU's on FORMS, namely, PTU61, PTU70, PTU91. It also contains corrections to the FORMS manual PDR3040.

New HELP files for FORMS, FDL and FAP are also supplied.

2 New Catalog Structure

The area causing most of the performance degradation in the existing FORMS system is the format of the catalog. In the updated FORMS a new catalog structure is implemented to improve the performance.

2.1 Detail



"fms.**" is used by pre-Rev19.3 FAP to maintain details of each form in the catalog.
"admcat" is used by Rev19.3 FAP to maintain details of each form in the catalog.
"lnk.fd" is the existing catalog and each form is stored with a filename of l.nnnn, where nnnn is a unique 4 digit number.
"pst100" etc. is the updated catalog, with a separate ufd for each device defined. Forms are stored under their own names in these ufd's.

"lnk.fd" and "fms.**" are no longer used, and may be deleted by the user after the conversion utility is run during the FORMS installation.

2.2 Multiple Catalogs

The FORMS* ufd is the default catalog ufd and must exist on each system. It is set up at FORMS install time.

Any other ufd on the system may also be used as a FORMS catalog. A catalog is created by issuing the PRIMOS create command, and then entering FAP and issuing a catalog and a create command.

NOTE: a user running an application in R-mode may only access a catalog which is a top level directory.

3 Build and Install Changes

3.1 General

Some extra modules are included in the build files.

The Vistar3 terminal driver is no longer automatically included in the build.

There is a new option, -no_convert, (or -nconv) on the install command line. This causes forms in the existing catalog structure NOT to be converted into the updated catalog structure. This is useful if forms have previously been converted to the updated catalog.

If the TCB option is used, which generates code to access the TCB information in the TCB* ufd rather than the FORMS* ufd, then only logical disk zero of the current system is searched for the TCB* ufd. This allows each system in a network to have a unique TCB directory.

3.2 Migration

At FORMS install time, a conversion utility is run to convert forms from the existing catalog format to the updated catalog format. This leaves the user with two sets of his forms, one usable by the existing FORMS system, and one usable by the updated FORMS system.

The existing FORMS and the updated FORMS may thus be run on the same system, as long as one is shared and one is non-shared. This is achieved by running the updated FORMS build file, but not subsequently running the updated FORMS install. The existing FORMS is left as shared and the non-shared versions are in the ufd LIB. The non-shared versions of the updated FORMS are in the ufd FORMS>LIB. An application programmer may take them from here in his load sequence.

examples:

V-mode

```
SEC
vload test.seq
lo    test
li forms>lib>nvforms
li vspool
li vapplb
li npftnlb
sa
o
```

R-mode

```
 LOAD
 lo test
 li forms>lib>rforms
 li spool$
 li applib
 li ftnlib
 sa test.save
 o
```

3.3 Loading

The following are examples of load sequences for R-mode, non-shared V-mode and shared V-mode. There are no changes to the shared V-mode load. The applications library must be included in the load of R-mode and non-shared V-mode.

R-mode
```
LOAD
lo test
li rforms          <---- R-mode FORMS library
li spool$
li applib          <---- NOTE  extra library required
li ftnlib          <---- R-mode FORTRAN library
sa test.save
q
```

NOTE: when running the Loader a warning message, ENTRY/COMMON NAME CONFLICT, ALWAYS appears when the RFORMS library is included. This is expected and should be ignored. This is due to code to initialise a Common area in a specific manner which doesn't worry SEG but does cause LOAD to issue a warning message.

NOTE also that space must be provided within an R-mode program for the forms being used. The following is an example for a FORTRAN program.

```
C    COMMON BLOCK FOR IOLIST
        PARAMETER IOLSIZ = 2500
        COMMON /IORCMA/ IBUF(3), IOL(IOLSIZ)
C
        DATA IBUF/IOLSIZ,0,0/
```

Non-shared V-mode
```
SEG
vload test.seg
lo test
li nvforms         <---- non-shared FORMS library
li vspool$
li vapplb          <---- NOTE   extra library required
li nvftnlb         <---- NOTE   li on its own is not correct
sa
q
```

Shared V-mode
```
SEG
vload test
mi
lo test
li vforms
li
```

sa
q

The loading sequence for shared V-mode is unchanged between the existing and the updated FORMS.

4 FDL Changes

There have been no enhancements made to FDL. Some inconsistencies in the command line handling have been corrected, and the compiler now processes listing options in a similar manner to the other PRIME translators.

In practice this means that if listing_no is specified, a listing never appears.

5 FAP Changes

FAP now has the ability to work on FORMS catalogs other than FORMS*.

The syntax of the new catalog command is as follows:
CATALOG [<pathname>]

<pathname> may be any valid PRIMOS pathname up to a maximum of 128 characters. CATALOG may be abbreviated to CAT.

To work on another catalog the user issues a catalog command with the name of the required catalog. The other FAP commands then refer to that catalog rather than FORMS*.

FAP starts with FORMS* as the default catalog. The command CATALOG on its own reverts to FORMS*. CATALOG fred looks for a top-level ufd of that name. CATALOG *>fred>bloggs looks for a subufd of that name relative to the HOME ufd (i.e. the ufd FAP was run from). Creating a new user catalog involves issuing the PRIMOS create command, then going into FAP and issuing CATALOG and CREATE. Only the FMS.** seqfile and the device ufds exist in user created catalogs. References to tcp.pm, dcf files, devext, devip etc. all still refer to FORMS*.

A quit handler has been added to FAP. All files are now closed if control-p is depressed while FAP is running.

6 Run Time Directive Changes

6.1 Terminator

Runtime directives from applications may now terminate in an exclamation mark. This will prevent runtimes reading through a complete input buffer when only the first few characters are significant, thus speeding up the interpretation of directives. Directives without terminating exclamation marks will continue to be interpreted as before. If more than one directive is given in the same line, then the exclamation mark may follow any in the list. Fortran applications return the correct length of the directive to the runtimes, and so no speed increase should be measured here. Cobol however returns the length of the buffer, which may be far greater than the length of the directive in the buffer, and thus speed increases should be expected when the directive terminator is used in Cobol applications.

The following are examples of the use of the terminator in directive lines:

```
'##LOG-OPEN!'
'##INVOKE TEST1!'
'##CLEAR ALL'/'##RELEASE!'
'##CLEAR ALL!'/'##RELEASE!'          ##RELEASE is NOT ignored
```

6.2 ##print

The syntax of the new ##print command is as follows:

```
     ##PRINT [<name>] [AT <dest>] [COPIES <number_of_copies>]
                      [FORM <form_type>] [NOHEADER]
  OR
     ##PRINT LOCAL
```

where an item in square brackets is an option, and items surrounded by '<', '>' are user supplied variables.
FORM and FORMS are equivalent, as are NOHEADER, NOHEAD and NOH.
Notice that the directives are not preceded with a minus sign warning character. This is deliberate, and is done to keep compatibility with the existing format of parameters to runtime directives.

The following spool options are not appropriate for ##print:

AS	CANCEL
DEFER	FTN
FUNIT	LIST
LNUM	NO_FORMAT
OPEN	PLOT
TUNIT	

A name may be specified after the ##print command, and this is used as an alias for the spooled file. This name is limited to 8 characters (ie anything after the 8th character is ignored).

##print LOCAL works as before, and takes no parameters. A FORMS runtime error occurs if parameters are supplied. LOCAL may be in upper or lower case.

The following are examples of the ##print options:

```
'##PRINT'
'##PRINT TEST1'
'##PRINT AT DOC'
'##PRINT TEST1 AT DOC'
'##PRINT NOHEADER'
'##PRINT TEST FORMS PLOT COPIES 4'
'##PRINT LOCAL'
```

6.3 The ##log Commands.

A new set of commands, ##log-open, ##log, and ##log-close are supplied. These commands allow the user to group the FORMS output from the particular application into a single spool listing.

6.3.1 ##log-open.

The ##log-open command opens a file on the spool queue, ready for the ##log commands that will write pictures of the forms to this file. No more than one log file may be open at any one time.

The syntax of the ##log-open runtime directive is as follows:

```
##LOG-OPEN [<name>] [spooler options, as ##print]
                   [DEFER <time>]
```

Note that the DEFER option is supported by ##log-open, but the LOCAL option is not. Also note that if DEFER is specified then the format that the time may take is restricted when compared with spooler; <time> ::= hh:mm, always in a 24 hour format. Leading zeroes in the hours field may be omitted, but must be present in the minutes field.

6.3.2 ##log.

This runtime directive appends a picture of the currently invoked form onto the spool file opened by the ##log-open command. ##log takes no parameters, and generates a FORMS runtime error should there not be a log file open.

6.3.3 ##log-close.

This runtime directive, which takes no parameters, closes and spools the log file, which must have been opened previously by a ##log-open command.

The following is an example of the use of the ##log commands.

```
      INTEGER*4 NAME
      CALL FORM$I
      WRITE (1, 5)
5     FORMAT ('##LOG-OPEN')
      WRITE (1, 10)
10    FORMAT ('##INVOKE SDI1')
      READ (1, 20)NAME
20    FORMAT (2A2)
25    CONTINUE
C         Relevent processing here: clear typed-in fields,
C                                   read in from the form etc
      WRITE (1, 30)
30    FORMAT ('##LOG')
      IF (MORE) GOTO 25
      WRITE (1, 40)
40    FORMAT ('##CLEAR ALL'/'##RELEASE')
      WRITE (1, 50)
50    FORMAT ('##LOG-CLOSE')
      STOP
      END
```

6.4 ##catalog

A new command ##catalog is provided to define the catalog on which subsequent directives operate.

The syntax of the new ##catalog command is as follows:

```
##CATALOG [<pathname>]
```

Pathname is any valid PRIMOS pathname up to a maximum of 128 characters. ##catalog without a pathname causes the catalog to revert to FORMS*. The runtime system starts off with FORMS* as the default catalog.

7 Compatibility Issues

Five areas must be considered when discussing compatibility
- defining forms
- the catalog structure
- the run-time directives
- search rules
- migration to the updated FORMS.

7.1 Forms Definition

A small change has been made to fix a bug, but no other changes have taken place in the FDL area, and there is no compatibility problem. See section 4 for details.

7.2 Catalog Structure

The detail of the catalog structure has changed. Within a catalog there is a sub-ufd for each terminal type and forms are stored within these sub-ufd's. A utility is provided, to run at installation time, which converts the old catalog structure to the new structure. FAP has changed internally to handle the new structure. The catalog structure is detailed in section 2.1, and the conversion utility is detailed in section 3.2.

The system allows multiple FORMS catalogs and an application may specify which one it uses. The default is FORMS* as at present. A new FAP command is provided to accept a filename for the catalog.

7.3 Run-time Directives

There is no compatibility problem with the existing run-time directives. All existing applications using shared FORMS will run unchanged with the updated FORMS. Applications loaded using the non-shared V-mode or R-mode FORMS libraries must of course be reloaded to include the updated FORMS. This is detailed in section 3.3

New options are available on the *#print directive, new directives are available, and all directive lines may be terminated with an exclamation mark. This is detailed in section 6.

7.4 Search Rules

The rules for searching for the FORMS catalog and the TCB list have changed.

The FORMS catalog may be anywhere on the system. Locally connected disks are searched first, and if the catalog is not

found, remote disks are then searched. If the TCB option is not included at build time, the file TCB.BN within the ufd FORMS* is used to ascertain the type of terminal on the line. If the TCB option is included at build time, the install routine creates TCB* as a top level ufd on logical device zero. Disks on remote machines are no longer searched for TCB*. If this ufd is not found by the runtime routines, the file TCB.BN in FORMS* is accessed instead.

7.5 Migration

It is quite feasible to run the existing and the updated FORMS systems at the same time and on the same system if required. This is detailed is section 3.2.

Subject: FSULIB

Release: 19.3

Date: 11/4/83

1 Description

This is the place for common subroutines used by the futil replacement commands. It is not being released for general user use. The calling sequences and functions performed can be changed without notice. If any of the subroutines ever get released on their own then they will be documented and supported like any other released product. The routines here are supported as if they were part of the futil replacement commands; which are LD. COPY, RWLOCK, PROTECT, and DELETE.

Included in the build is the building of the file FSULIB>BIND.SAVE which in turn is used to load the commands.

2 Environment

Needs PRIMOS 19.0.65 or greater. (19.0.respin)

3 Installation and Build Procedures

Needs to be run before building the following products: LD, COPY, RWLOCK. PROTECT, and DELETE.

4 Problems Fixed

Fixed a bug in VFY$ that causes VFY$ to call ERRPR$ with a key of K$NRTN which is not allowed in an EPF. Fixed a bug in CP$DIR that causes incorrect error due to simple name instead of treename in attribute copying.

5 Problems Fixed at 19.3

Made changes to handle the case of deleting delete_protected directories and answering "no" to the verification question. Previously this would return no error message, changes were incorporated in this fix so that now an error message will be printed.

Subject: FTN

Release: 19.3

Date: November 4, 1983

1 New Functionality

None.

2 Problems Fixed -- User-reported

2.1 SPAR 3000096

A problem was fixed where the compiler emitted incorrect code for indexing into a certain multiply-dimensioned array inside a DO-loop, when the program was compiled with the "-OPT" option.

3 Environment

Requires PRIMOS Rev 19 to run. Requires Rev 19 master disk to build.

4 Installation and Build Procedures

Standard.

File Transfer Service (FTS).

|Release Number 2.0

|Software Issue number 19.

|Primos Revision Number 19.3

|Date : 2nd September '83

Functionality

The functionality of FTS is documented by Technical Publications in the 'Prime User's Guide', 'System Operator's Guide', 'System Administrator's Guide'. and the 'Primenet manual'.

|New functionality provided by FTS Rev. 2.0 :-

|The data transfer throughput of the FTS servers has been significantly |improved.

|The idling memory working set of the FTS manager process (YTSMAN) and |the FTS server(s) has been reduced considerably.

|The FTS server and FTR utility are now shared programs.

|Standard SEGSAM or SEGDAM - type segment directories can be transferred |between Rev. 2.0 FTS sites.
|So, V-mode runfiles and MIDAS files can now be transferred by FTS.

|A new FTR Request Management option (-STATUS_ALL) has been implemented. |This allows users to display the status of ALL user's requests, not |just their own requests.

|The FTR Request Management options have been enhanced to allow the user |to specify the desired action to be performed on all his/her requests.

|The FTR Submit options -DSTN_FILE_TYPE and -SRC_FILE_TYPE can be used |to specify the expected or required type of the source/destination |file.

|When submitting a request and the local file is greater than 250K |bytes, the user will be queried to see if a temporary copy of the file |is to be made.
|A -NO_QUERY submit option is provided to suppress such queries.
|The default is to query the user.

|The numbering of requests has been enhanced to be a unique number of |the form 'nnnn', where 'nnnn' is a four digit number with leading |zeroes suppressed.
|Requests will be assigned the next available sequential number when |they are submitted.

|The display produced by the FTOP -LIST_SRVR_STS command has been |enhanced to include the local treename/device involved in the transfer, |the time the transfer started, a send/fetch indicator, and the remote |site name.

|A new FTOP option (-STOP_MNGR) allows the operator to close down the |FTS manager process , YTSMAN.

|A new FTGEN site subcommand (ISSUE) allows the issue no. (version) of |the FTS system on a remote site to be configured.

|Server phantom events have been included in the server log file.

|The subroutine interface to FTS has been documented and released.
|The subroutine name is FTISUB .

|The matching of recieved server passwords and that contained in the FTS |configuration database has been made case-INSENSITIVE.

|Server passwords and site addresses are stored in the FTS configuration |database in an encrypted form.

|All FTS commands and programs will display the revision number 2.0 , |instead of 1.x .

|Problems Fixed. - (Compared with Rev. 19.1, Issue 17).

User visible.

|Previously the FTGEN server subcommand PRIORITY would reset the |priority to zero when a textual value was specified. Specifying a |textual value is now errored and the configured priority value is left |unchanged.

|The log file RESULT message of "File Transfer Protocol Error" has been |replaced with various text messages to indicate clearly the cause of |the transfer failure.

|The FTGEN command to delete a site (DELETE_SITE) would crash with an |ACCESS_VIOLATION if the site did not exist in the FTS configuration.
|Attempting to delete a non-existant site now causes a "not configued" |error message to be produced.

|Previously, a FTR -ABORT of a waiting request caused the request to be |deleted from it's request queue.
|An FTR -ABORT of a request in progress would result in the request |being held by FTS.
|Changes have been made so that an FTR -ABORT of a waiting or active |request results in the request being held.

|The FTGEN PURGE_QUEUE command failed to delete all requests from the |specified queue.
|This problem has been fixed.

|FTR will no longer uppercase the open-network address form of |-DSTN_SITE and -SRC_SITE submit options.
|This allows the server password to be specified in lower case if |necessary.

|Previously, when the FTS database was invalid the LSITE command would |return a 'Not found' error message.
|Fixed to consistently return a message of 'The FTS database is |invalid'.

|The FTR -DISPLAY command now consistently displays request block |information in uppercase.

|When a request fails and is put on hold, the RESULT and |Source/Destination notify message has been improved to make it clear |that the request will NOT be retried.

|When an FTS server phantom starts up, it now displays the FTS revision |number. This is consistent with all other FTS programs.

|Various FTGEN commands expect filenames as their arguments.
|Previously, these commands allowed the user to specify illegal |filenames.
|FTGEN now validates that all filenames conform to the Prime standard.

|Corrected the problem whereby a space character did not appear between |the error text and error token for FTS error messages.

|If an FTS server is started up without it's associated queue being |configured, the abort message produced has been enhanced to read :-
|'Server closed down - Server queue not configured.'
|rather than :-
|'Server closed down - Error reading server queue - illegal name'.

|FIX for SPAR Id. 2006268, or POLER No. 61992.
|Rev. 1.x versions of FTS caused the destination file to be created as |a DAM - type file, if the destination file did not exist.
|This was done irrespective of whether the source file was a SAM or DAM |- type file.
|For transfers between Rev. 2.x sites the destination file type will be |the same as the source file.
|For transfers between Rev. 1.x and Rev. 2.x FTS sites, the |destination file is still created as a DAM - type file if the |destination file does not already exist.

|The FTR command line is uppercased to conform to the Command Argument |Standard.

Not user visible.

|Changes to conform to the revised Master Disk standard.
|Changed all Build and Install files to use COPY and DELETE instead of
|FUTIL.
|Useo PWLOCK PRIMOS-level command instead of FUTIL's SWRLOC command.
|All HELP files have a '.HELP' suffix.
|Changed all Build files to catch 'WARNINGS' as well as 'ERRORS'.
|Ensured that Build files echo PRIMOS commands.

|Corrected all build files so that MAP commands are positioned after the
|SAVE command.
|This means that the load maps will show the true stack position, rather
|than a null pointer.

|FTSSRC>LIBRARY>SOURCE modules which were only used by one component of
|FTS, have been relocated in the appropriate component SOURCE sub-ufd.
|Only genuinely common routines have been included in the combined QPAKS
|and FTS library shared segments.

Outstanding Problems

|There are no outstanding SPAR's.

Environment

|FTS Rev. 2.0 will run on all Rev. 19 PRIMOS revisions.
|This issue of FTS (Issue 19) will NOT run on PRIMOS rev. 18.2 and
|beyond in the 18.x range.
|Issue 19 makes use of the SWSINT subroutine, which is only available at
Rev. 19.0 and beyond.
|In addition, a named semaphore is used by YTSMAN to allow FTOP manager
|commands to control the start up and shut down of YTSMAN. The file
|system object associated with this named semaphore is FTS0*>YTSMAN.COMI
|.

|To build Rev. 2.0. FTS requires a 19.3 system.
|More specifically, the 19.3 version of SYSCOM>X$KEYS.INS.PL1 and
|X$KEYS.INS.FTN .

FTS does not use any segments beyond 4007, in the 4000 range.
|FTS uses the following allocated segments :-

| 2026
| 2027
| 2126
| 2127
| 6001 - Locations 0 to 32777.

FTS utilises the following PRIMENET port numbers :-

Ports 252 and 256 are utilised by the FTS manager phantom, YTSMAN, when processing incoming requests.

For every configured FTS server, a port number is required in the range 1 to 99. Each server requires it's own unique port number, which must not be used by any other process on the machine.

FTS Build Procedures

The command to build the complete FTS subsystem is :

```
resume ftssrc>fts.build.cpl [-como]
```

The -como option will generate a comoutput file called fts.build.como.

The command to spool the listings of the complete FTS subsystem is :

```
resume ftssrc>fts.list.cpl
```

Individual FTS modules may be built and listed by the appropriate cpl files in the ftssrc directory.

If there are requirements to do anything more complicated, then it is suggested that you refer to the cpl build file listings as there exist a number of other options that would normally only be used in a development environment that may be useful.

FTS Installation Procedures

Having built FTS as indicated above, it must now be installed by following the appropriate course of action from those detailed below.

Note - FTS0* is the run time directory for FTS and as well as containing the FTS database, also acts as the spool directory for the FTS transfer requests.

Initial installation. (No previous FTS system installed).

I.e when no FTS0* directory exists at the MFD level.

Run the command file :-

```
cominput fts>fts.initinstall.comi
```

This will create the FTS0* directory at the MFD level and copy up the contents of the FTS>FTS0* directory.
In addition :-

The FTS utilities FTR, FTOP, FTGEN are copied to CMDNC0 .

The shared segments FQ2026, FQ4000, FQ2126A, FQ2126B, SV2127, FR2126A, FR2126B, and the command file, FTS.SHARE.COMI, are copied to the SYSTEM directory.

The files FTSSUB.INS.PL1 and FTSSUB.INS.FTN are copied to the SYSCOM directory.

The customer-loadable library VFTSLB is copied to the LIB directory.

NOTE - fts.initinstall.comi may need modifying to specify explicitly the particular command partition where FTS0* is to reside.

If FTS0* is an ACL protected directory, then you must ensure that YTSMAN and the file transfer server(s) (e.g. FTS1) are granted ALL ACL access rights to the FTS0* directory. Otherwise, the YTSMAN and file transfer server phantoms will fail to start up.
In addition, the user SYSTEM needs ALL access rights to FTS0* and the $PEST access rights should be set to DALURW .

Edit the file CMDNC0>C_PRM0 to include the following :-

(i) A command to share FTS for use

```
cominput system>fts.share.comi 10
```

This command should be inserted AFTER initialisation of the SPL run-time library, and BEFORE the following two commands.

(ii) A command to phantom the file transfer service manager

```
ftop -start_mnor
```

(iii) A command to phantom the file transfer server

```
ftop -start_srvr fts1
```

Note.
The "ftop" commands should be inserted in C_PRMO after the command to set the system date and time.

From the system console, run the command file :-

```
cominout system>fts.share.comi
```

Now invoke FTGEN and configure the local FTS subsystem. Note that you MUST be logged in as SYSTEM in order to use FTGEN, and that a FTGEN "INITIALIZE_FTS" command MUST be performed before configuring the system.

After FTS has been installed on sites that have X.MAIL, you should create the file "MAIL_ON.FTS" in the FTS0* directory. This will enable users to send mail via FTS and to recieve source/destination notify messages via X.MAIL .

Following configuration, perform a cold start to test out the C_PRMO edits and to start up the YTSMAN and file transfer server phantom(s).

A simple example configuration sequence is shown below :

```
OK, ftgen
[FTGEN rev 2.0]
FTS STATUS
----------
Server directory is ftsq*.
System issue number is  1?.
The FTS data base is invalid.  (status)
ftgen> initialize_fts          /* Initialize the new system.
FTS STATUS
----------
Server directory is ftsq*.
System issue number is  19.
Number of queues configured is  0.
Number of servers configured is  0.
Number of sites configured is  0.
ftgen> add_queue fts%1         /* Add a single queue.
queue: log -off
queue: maximum_requests 200
queue: file
Queue added.
ftgen> add_server fts1         /* Add a single server.
server: queue fts%1
server: log fts1.log
server: message_level detailed
server: port 2
server: file
Server added.
ftgen> add_site loc1           /* Add the local site.
site: address loc1+fts1
site: issue 19
site: queue fts%1
site: log -off
site: file
Site added.
ftgen> add_site rem1           /* Add a remote Rev. 2.0 (Issue 19) site.
site: address rem1+fts1
site: issue 19
site: queue fts%1
site: log rem1.loc
site: message_level detailed
site: file
Site added.
ftgen> add_site rem2           /* Add a remote Rev. 1.1 (Issue 17) site.
site: address rem2+fts1
site: issue 17
site: queue fts%1
site: log rem2.log
site: message_level detailed
site: file
Site added.
ftgen> quit          /* OK that's it, let's quit.
OK,
```

Re-installation of Rev. 2.0 onto existing Rev. 2.0 system.

The recommended procedure is to allow any outstanding queued or transferring file transfer requests to be completed, before attempting to re-install. The FTP server(s) should be closed down (FTOP -STOP_SPVR <server_name>).
The FTS manager, YTSMAN, should be closed down (FTOP -STOP_MNGR).

It is a good idea to take a back up copy of the current FTSQ* UFD and the FTS utilities in CMDNC0, SYSTEM, SYSCOM, and LIB, just in case of any problems. The previous version can then be easily re-instated.

Firstly, run the command file :-

```
cominput fts>fts.install.comi
```

This copies the contents of the FTS>FTSQ* directory into the FTS0* UFD, replacing matching files that already exist in the FTSQ* UFD.
In addition :-

- The FTS utilities FTR, FTOP, FTGEN are copied to CMDNC0 .
- The shared segments FQ2026, FQ4000, FQ2126A, FQ2126B, SV2127, FP2126A, FP2126B, and the command file, FTS.SHARE.COMI, are copied to the SYSTEM directory.
- The files FT$SUB.INS.PL1 and FT$SUB.INS.FTN are copied to the SYSCOM directory.
- The customer-loadable library VFTSLB is copied to the LIB directory.

If FTSQ* is an ACL protected directory, ensure that YTSMAN and the file transfer server(s) (e.g. FTS1) are granted ALL ACL access rights to the FTS0* directory.
In addition, the user SYSTEM needs ALL access rights to FTSQ* and the $REST access rights should be set to DALURW .

Ensure the file CMDNC0>C_PRM0 includes the following :-

(i) A command to share FTS for use

```
cominput system>fts.share.comi 10
```

This command should be inserted AFTER initialisation of the SPL run-time library, and BEFORE the following two commands.

(ii) A command to phantom the transport service manager

```
ftop -start_mngr
```

(iii) A command to phantom the file transfer server

```
ftop -start_srvr fts1
```

Note.
The 'ftop' commands should be inserted in C_PRMO after the command to set the system date and time.

From the system console, run the command file :-

```
cominput system>fts.share.comi
```

Now invoke FTGEN and do an 'initialize_fts' (abbreviation 'ifts'), and check over the configuration.

For sites where X.MAIL is installed, check that a file 'MAIL_ON.FTS' file is present in the FTS>* directory. If this file exists FTS will utilise X.MAIL when sending source and destination notification messages, thus allow mail to be transferred using FTS.

Following all this, EITHER perform a cold start to check out any C_PRMO changes and start up the new YTSMAN and file transfer server phantoms(s), OR start up the manager and server(s) using the above FTOP commands from the system console.

Installation of Rev. 2.0 on existing Rev. 1.x sites.

IMPORTANT NOTE.
The Rev. 2.0 system CANNOT be simply installed on top of a Rev. 1.x FTS system for the following reasons :-

- Rev. 1.x requests CANNOT be processed by Rev. 2.0 FTS, and vice-versa.

- Rev. 1.x configuration formats are incompatible with Rev. 2.0 and vice-versa.

So it is IMPORTANT to perform the install in the manner outlined below :-

Use the FTGEN BLOCK_QUEUE command to prevent users from submitting further transfer requests whilst the install is in progress.

Since Rev. 1.x requests will NOT be processed by Rev. 2.0, it is advisable to ensure all outstanding requests have been transferred and removed from the queue(s) before commencing installation.
Alternatively, make a COMOUTPUT record of the outstanding requests using a FTR -DISPLAY command whilst logged in as SYSTEM. This record can then be used to re-submit the requests following installation.

The FTP server(s) should be closed down (FTOP -STOP_SRVR <server_name>).
The FTS manager, YTSMAN, should be closed down (FTOP -STOP_MNGR).

Make a COMOUTPUT record of the Rev. 1.x configuration using the FTGEN commands, LS -ALL, LSITE -ALL, and LP -ALL .
This is essential, since following installation, you will need to re-configure the FTS configuration.
This COMO file can also be edited to form a COMINPUT file to reduce the amount of re-keying of configuration information and reduce the risk of errors.

Preserve the current Rev. 1.x software just in case of any problems. Namely :-

- Take copies of the SYSTEM files FTS.SHARE.COMI, 0400C, and QP2026 .

- Take copies of the CMDNCO files FTR.SAVE, FTOP.SAVE, FTGEN.SAVE .

- CNAME the FTS0* to FTS0*_SAVE, for example.

Firstly, run the command file :-

```
cominput fts>fts.initinstall.comi
```

This will create the FTS0* directory at the MFD level and copy up the contents of the FTS>FTS0* directory.

In addition :-

The FTS utilities FTR, FTOP, FTGEN are copied to CMDNC0 .

The shared segments FQ2C26, FQ4000, FQ2126A, FQ2126B, SV2127, FR2126A, FR2126B, and the command file, FTS.SHARE.COMI, are copied to the SYSTEM directory.

The files FT$SUB.INS.PLI and FT$SUB.INS.FTN are copied to the SYSCOM directory.

The customer-loadable library VFTSLB is copied to the LIB directory.

NOTE - fts.initinstall.comi may need modifying to specify explicitly the particular command partition where FTSQ* is to reside.

If FTSQ* is an ACL protected directory, ensure that YTSMAN and the file transfer server(s) (e.g. FTS1) are granted ALL ACL access rights to the FTSQ* directory. Otherwise, the YTSMAN and file transfer phantoms will fail to start up.
In addition, the user SYSTEM needs ALL access rights to FTSQ* and the $REST access rights should be set to DALURW .

Ensure the file CMDNC0>C_PRMQ includes the following :-

(i) A command to share FTS for use

```
comiput system>fts.share.comi 10
```

This command should be inserted AFTER initialisation of the SPL run-time library, and BEFORE the following two commands.

(ii) A command to phantom the transport service manager

```
ftop -start_mngr
```

(iii) A command to phantom the file transfer server

```
ftop -start_srvr fts1
```

Note.
The 'ftop' commands should be inserted in C_PRMQ after the command to set the system date and time.

From the system console, run the command file :-

```
comirout system>fts.share.comi
```

Now invoke FTGEN and re-configure the FTS configuration.
Pay particular attention to the setting of the new site configuration parameter, as specified by the ISSUE site subcommand.

|After FTS has been installed on sites that have X.MAIL, you should |create the file 'MAIL_ON.FTS' in the FTSO* directory. This will enable |users to send mail via FTS and to recieve source/destination notify |messages via X.MAIL .

|Following all this, FITHER perform a cold start to check out any C_PRMO |changes and start up the new YTSMAN and file transfer server |phantoms(s), OR start up the manager and server(s) using the above FTOP |commands from the system console.

|Finally, re-submit any outstanding Rev. 1.x requests using the |COMOUTPUT record made prior to installation.

Recovering from a corrupt FTS database.

The FTSQ* directory is the run-time directory for FTS, and as well as containing the FTS configuration database, also acts as the spool directory for the FTS transfer requests.
One of the functions performed by FTS is validation of the FTS database. Whilst the database is valid, the file VALID_SUBSYTEM.FTS will exist in the FTSQ* directory. Should invalid data be detected, the VALID_SUBSYTEM.FTS file is deleted, and the database invalidated. The likelihood of this event occurring is very small, but the following is a recover strategy, in the order it should be attempted. Only proceed to the next stage if the preceeding one fails.

- Invoke the FTGEN utility and try an INITIALIZE_FTS. Check the configuration is complete using the FTGEN commands LSITE -ALL, LS -ALL, and LQ -ALL .

- Close down the file transfer servers, using the FTOP utility, and then logout the YTSMAN phantom.

  Perform an initialization of the FTS subsystem by:

  ```
  RESUME FTSQ*>INIT -WORLD_RESET
  ```

  Followed by :

  ```
  RESUME FTSQ*>HINIT
  ```

  Do an INITIALIZE_FTS in FTGEN to see if the database is now valid.

- The ultimate solution is to attach to the FTSQ* directory and delete all the files with a '*.FTS' suffix.

  Then perform a:

  ```
  RESUME FTSQ*>INIT -RESET -SUBSYSTEM FTSQ*
  ```

  Do an INITIALIZE_FTS.

  Re-configure the FTS configuration of queues, sites and servers using FTGEN.
  For sites with X.MAIL, the file MAIL_ON.FTS will need re-creating.

Structure of FTS UFD's

```
FTSSRC        /* The top level FILE TRANSFER SERVICE source directory.
                 This directory contains all the source of the FTS plus
                 the command files to Build and List the complete
                 FTS system. */
 FTR          /* The FTR Interactive submittal program directory. */
  SOURCE      /* The directory that contains the FTR source. */
  BINARY      /* The directory that contains the FTR binary files. */
  LISTING     /* The directory that contains the FTR listing files. */
  INSERT      /* The directory that contains the FTR insert files. */
 FTGEN        /* The FTGEN utility directory. */
  SOURCE      /* The directory that contains the FTGEN source. */
  BINARY      /* The directory that contains the FTGEN binary files. */
  LISTING     /* The directory that contains the FTGEN listing files. */
  INSERT      /* The directory that contains the FTGEN insert files. */
 FTOP         /* The FTOP utility directory. */
  SOURCE      /* The directory that contains the FTOP source. */
  BINARY      /* The directory that contains the FTOP binary files. */
  LISTING     /* the directory that contains the FTOP listing files. */
  INSERT      /* The directory that contains the FTOP insert files. */
 FTP          /* The FTP server program directory. */
  SOURCE      /* The directory that contains the FTP source. */
  BINARY      /* The directory that contains the FTP binary files. */
  LISTING     /* the directory that contains the FTP listing files. */
  INSERT      /* The directory that contains the FTP insert files. */
 LIBRARY      /* The FTS LIBRARY directory (FTS common subroutines). */
  SOURCE      /* The directory that contains the LIBRARY source. */
  BINARY      /* The directory that contains
                 the LIBRARY binary files. */
  LISTING     /* The directory that contains
                 the LIBRARY listing files. */
  INSERT      /* The directory that contains
                 the LIBRARY insert files. */
 SYSCOM       /* The directory containing the source of files which
                 will be placed in SYSCOM top-level UFD. */
 YTS          /* The directory containing the source of the
                 Yellow book Transport Service. */
  SOURCE      /* The directory that contains the YTS source. */
  BINARY      /* The directory that contains the YTS binary files. */
  LISTING     /* The directory that contains the YTS listing files. */
  INSERT      /* The directory that contains the YTS insert files. */
| YTSMAN       /* The directory containing the source of the
|                 FTS Manager process, YTSMAN. */
|  SOURCE     /* The directory that contains the YTSMAN source. */
|  BINARY     /* The directory that contains the YTSMAN binary files. */
|  LISTING    /* The directory that contains the YTSMAN listing files.*/
|  INSERT     /* The directory that contains the YTSMAN insert files. */
 QPAKS        /* The directory that contains the source for the
                 QPAKS queuing utility. */
  SOURCE      /* The directory that contains the QPAKS source. */
  INSERT      /* The directory that contains the QPAKS insert files. */
  UTILS       /* The directory that contains utility files for
```

```
                building CPAKS. */
   LIB          /* The directory that contains the CPAKS library
                   runfile. */
 | HINIT        /* The directory containing the source of the
 |                 FTS Histogram initialisation program. */
 |  SOURCE      /* The directory that contains the HINIT source. */
 |  BINARY      /* The directory that contains the HINIT binary files. */
 |  LISTING     /* The directory that contains the HINIT listing files. */
 |  INSERT      /* The directory that contains the HINIT insert files. */

 FTS            /* This directory contains all the FTS run files,
                   installation command files, a share command file,
                   and a skeleton of the FTSQ* directory. */
  CMDNCO        /* Contains the run files for the FTR,FTOP,
                   and FTGEN utilities. */
  FTSQ*         /* Contains the skeleton of the FTSQ* directory. */
  INFO          /* Contains FTS build and installation instructions
                   and details relating to the current release. */
 | LIB          /* Contains the VFTSLB library which is copied up
 |                 to the LIB directory. */
 | SYSCOM       /* The directory containing a copy of files which will
 |                 be placed in SYSCOM top-level UFD. */
  SYSTEM        /* Contains the FTS shared segments which are copied
                   to the SYSTEM directory. */

 FTSQ*          /* This directory contains the FTS configuration files
                   and all the FTS run-time files. */
  HELP.FTS      /* Contains all the HELP information
                   for the FTS system. */
   FTP          /* Contains the HELP information files
                   for the FTP utility. */
   FTOP         /* Contains the HELP information files
                   for the FTOP utility. */
   FTGEN        /* Contains the HELP information files
                   for the FTGEN utility. */

  COMO.FTS      /* Contains the Comoutput files associated
                   with File Transfer Servers. */
  LOCK.FTS      /* Contains the Lock files associated with
                   File Transfer Servers. */
  START.FTS     /* Contains files associated with the start up of
                   File Transfer Servers. */
```

*** END OF FTS.PUNO ***

/* ICS1.RUNI, ICS1SRC , hardware communications group, 09/02/83
/* irs1 z80 Info file
/* copyright (c) 1981, prime computer, inc., natick, ma 01760
/*
Subject: ICS1 (Primenet support)
Release: 19.3
Date: September 1,1983

1. New Functionality:
The ICS1 will now support Primenet X.25. This support is a level 1 (physical layer) support.

2. Problems fixed:
Added Primenet support to the Lynx.

Changed CLOK$ to allow a timer that was set to expire.

Added code to IPC$ to allow decueue from high priority queue only. Added code to allow rescheduling if a call to DMQPUT$ failed. Added code to disable interrupt during call to IPCENQL$.

Changed LC1 process from level 4 to level 2.

Changed IHST$ to set bit in OLDSTAT variable to allow processing of lost notifies in PIOPH$ (This corrects SPAR 3001718).

Changed PIOPH$ to allow processing of lost notifies. (This corrects SPAR 3001718).

Deleted module PIODAT$ because it contained second copy of data base variables.

Changed LCQ$ to allow the PRIMENET process to run.

Inititialed backplane DMA chip in INIT*L.

Allowed PRIMENET interrupts to run via IVS*.

Added PRIMENET process to HOSTLCC table.

Enlarged LC1 queues in QUEUES and QUEUES.INS.

Added routine FDQLOOK$ to allow the first member to be inspected without changing the pointers.

Added routine TSTQW$ to test if a queue is full, and to get the number of bytes used in the queue.

Added SCHED$ (RESCHED$) to allow a process to reschedule itself.

Changed DISP$ to allow for rescheduling a process.

Made change in ASYNC>COMMAND to allow baud rate generator fix. Made

chance in ASYNC>COMMAND to CMD9 portion that allows DTR to stay dropped when the "drop DTR" command is issued. (Corrects SPAR 2006161)

Made change in ASYNC>COMINIT to initialize the odd port of SIO chips before initializing the even port to prevent the Ics1 from vectoring to an incorrect address if there is a noisy line.

Deleted some test code that was left in the Bisync software.

3. Outstanding Problems:
The ICS1 can run with at most 4 virtual connections over the Synchronous X.25 line. Any more than 4 lines could cause the ICS1/PRIMOS interface to break down.

4. Environment:
This product cannot run with any previous releases of PRIMOS.

Date: 11/12/83
Subject: Bug fixes for Rev. 19.3.0

The latest fixes correct problems with handling end of range on Primos queues due to status errors (framing, parity errors), and eliminate out of order data, as well as a run-time self-verify problem when a LAC returns its ID but all four lines verify as bad, and a console monitor problem with dump (D) and search (S*) commands with no parameters.

The latest fix correct a problem with the handling of the line configuration command (command 0) from the ICS DIM, which caused DTP to be dropped improperly.

INDEX is a directory which holds tools for use in building Master Disk Software. It is not intended for customer use.

INTCOM* is a directory which holds insert files used in the build of Prime Master Disk software. It is not intended for customer use.

Changes have been made to the build file to detect error and warnings during the build. Also the rev stamp has been changed.

Subject: LD

Release: 19.3

Date: November 4, 1983

1 Changes made to LD at 19.3

2 New functionality

o The output of LD now has a new option called TRUNCATED. If a file has been truncated by FIX_DISK then under truncated 'yes' will be displayed, if not then 'no' will be displayed.

3 Internal bug fixes

o Using LD with the options -DET and -CATP now works correctly.

Subject: LOGPRT

Release: 19.3

Date: November 29,1983

- Changed references from 4650 to 9950.
- Modified to interpret PRIMOS and network logging files.
- Updated messasges for LYNX PRIMENET.

Subject: MAGLIB

Release: REV19.3

Date: June 28, 1983

1 Enhancements

MAGLIB and MAGNET have been placed in one directory. This was done so that the build process would be less cumbersome(eg. Now only one set of INSERT files is needed).

2 Changes at 19.3

o T$READ and MT$RDIM have been modified to detect any. tape record discrepencies. o Added checks for bad logical record size.

Subject: MAGNET

Release: REV19.3

Date: June 28, 1983

1 Enhancements

o MAGNET and MAGLIB have been placed in the same submittal directory so that building these products would be less prone to errors.

o Added checks for bad logical record size.

Subject: MAGSAV.MAGRST

Release: 19.3

Date: November 4, 1983

MAGSAV and MAGRST for Rev19.3

All of the changes to MAGSAV and MAGRST are those included as fix revs of R19.2, which are documented below.

Enhancements at Rev 19.2.1

POLER 52789

MAGSAV now supports a new command, $TTY, in order to allow the user to enter a new tape unit from the terminal during the course of a save. This works even if the -TTY option is not selected on the command line, but the first tape unit number can be input from a cominput or CPL file. An example of the use of $TTY appears below :

```
[MAGSAV Rev. 19.2.1]
Tape unit (9 Trk): 0
Enter logical tape number: 1
Tape name: TEST
Date (MM DD YY):
Rev no:
Name or Command: $TTY
Name or Command: DIR
E-O-T has occurred, mount next tape
1 Recovered MT IO errors.
New Tape unit (9 Trk): 1
Name or Command: DIR
Name or Command: $R
```

It is not generally possible to predict whether a save will reach the end of the tape, and this new command enables the user to intervene if necessary. Previously, if an &DATA block within a CPL file invoked MAGSAV, an unexpected E-O-T would cause the program to take the wrong data as the new tape unit an the CPL program would eventually fail.

Enhancements at Rev 19.2.2

REF master files were not previously allowed to appear in top-level ufd's, but ROAM now permits them to be. MAGSAV and MAGRST have therefore been enhanced to handle them.

Subject: MAKE

Release: REV19.3

Date: November 4, 1983

1 Changes to MAKE

The utility MAKE has been changed to save any badspot information that is already on the disk. This is done by adding the following options to MAKE:

1. ~-NEW_DISK~;~-NEWDSK~: Making a new disk partition. This option will suppress the attempt to read a badspot file from the disk before anything else is done.

2. ~-COPY_BADSPOTS_BY_NAME~<partition>~;~-CPYNAM~: Copy badspots from disk named <partition>. This will get the badspot file from the named disk's badspot file. The named disk must be a different logical partition of the same physical disk (but this will not be checked).

3. ~-COPY_BADSPOTS_BY_DEVICE~<pdev>~;~-CPYDEV~: Copy badspots from physical device <pdev>. This will read the badspot file from the logical disk partition specified. It will be checked to be on the same physical disk as the logical partition that we are making.

o MAKE has been changed to allow the inclusion of badspots that are not part of the logical disk. This allows the badspot file that is in each logical disk of a physical disk to contain the same information. With the added options to MAKE to copy badspot files from other partitions, this becomes a requirement. This copying from (possibly other) partitions is to make it harder to lose badspot files.

o Make will attempt to read a badspot file from the disk before anything else is done unless the -NEW_DISK option has been specified. The information read from the disk will be added to any records specified by the user and to any bad records found in the verify pass.

SUBJECT: MIDASPLUS Release Documentation

RELEASE: Rev 19.3

DATE: December 1, 1983

1 NEW FUNCTIONALITY COMPARED TO REV 19.2

CREATK COUNT OPTION

A new option has been added to CREATK to count all the entries in a specified index. The new option, 'COUNT', differs from the existing USAGE option in that it actually reads through an index, verifying and counting each entry. It then displays the total number of valid entries found. The USAGE option reads the number of entries inserted and deleted since the last MPACK from the index description which is maintained by the online MIDASPLUS routines. In the event of an abnormal exit from MIDASPLUS an index descriptor may be left in an erroneous state. Should this happen the COUNT option can replace the USAGE option to get the correct state of the file.

If a count is requested on a primary index (index = 0), CREATK will be able to determine the number of records inserted or deleted since the last MPACK and will update the file descriptor values if they differ. This can be done because data records are only marked for deletion and are not physically deleted until the file is MPACKed. The inserted/deleted statistics cannot be recreated from a secondary index and if there is a difference between the total count and the values of the index descriptor no update can be made. In this case a message indicating that the file decriptor no longer matches the true state of the file is displayed. An MPACK would be required to re-establish the correct file statistics for display through the USAGE option.

In order for CREATK to get an accurate count it must have exclusive use of the file. This is implemented through the access controls in the same way that they are used by MPACK for exclusive access.

See Appendix B for an example session using the new CREATK COUNT option.

SPY - A NEW MIDASPLUS UTILITY

Midasplus has in memory information that is used and updated during runtime. This information includes a table of data record locks taken, system wide statistics on the performance and use of the system, system wide configurable parameters, and user specific configurable parameters. The function of this utility is to display all or parts of this information during the running of Midasplus.

This utility will be useful to "spy" on the workings of Midasplus.

The three catagories of information available are :

1. Record Locks.

2. Statistics,

3. System Configuration.

For statistics, SPY will display the internal values at an interval of n seconds for m intervals. where n and m are supplied by the user. This provides a view of what is happening as the system works. Each information catagory is further described in Appendix A.

MPACK - no more ACCESS

As a result of the new method for concurrent access control between the runtime library and the utilities. MPACK no longer has the ACCESS option as it is no longer needed.

2 USER-VISIBLE BUGS FIXED

SPAR 3000797

The MPLUSLP routine KX$PFC was using a shared area to hold the per user file unit. This resulted in errors messages such as 'Unit not open', or 'Not a segment directory' for VRPG applications or any application that called KX$PFC and was running multi-user. This problem has been corrected.

SPAR 3000792

Midasplus file unit management routine was occasionly mixing up file units of one file with another. This problem has been corrected.

P60318

Read only files are now supported by Midasplus runtime.

NOTE: If a user has LUR access rights to a file, then that user can use Midasplus runtime but can not use the Midasplus Utilites (CREATK, MPACK, MDUMP, MIDDEL, MBUILD).

P54290

Adding variable length records now works.

DEADBEATS

The delete counter is now correctly incremented for "deadbeats" and no longer prints any message regarding them when in debug.

P57851

MDUMP no longer puts the key into the data record on very long records.

Static_On_Unit_Messages

When in debug mode Midasplus no longer prints out the "Making Static On Unit" message. In addition the static-on-unit, when invoked, now displays a message only when clean up is done, unless in debug mode.

MDUMP_Display_Problem

MDUMP now correctly reports the key position for records larger than 16K words.

Offline_Sequence_Flag

The offline utilities now correctly return a sequence flag of 3 when they have successfully closed up the files.

P57655

Midasplus no longer checks for rev. level regression.

P70974

Deleted duplicate entries no longer reappear after MPACK.

3 NON-VISIBLE_INTERNAL_FIXES_AND_ENHANCEMENTS

Concurrent_File_Access_/_Midasplus_runtime_and_utilities

Concurrent file access method between the Midasplus utilities and runtime has been changed in response to P60318 (see above). The new method does not require write access to the file to run Midasplus applications (but realize that you must have write access to use the utilites). The new method uses the Primos read/write lock. When a utility wants access to a file it first changes the r/w lock to EXCLUSIVE access and then opens the file for writing, if the file open is successful then the utility has exclusive access. The utility then restores the file's original r/w lock when it is finished. If a user breaks out of the utility with CNTRL-P then a dynamic on-unit will restore the r/w lock. As a result of this change MPACK no longer has an ACCESS option to reset the access control lock in a file, as that

lock no longer exists.

NOTE: THIS METHOD DOES NOT PROVIDE PROTECTION AGAINST UNSHARED MIDAS (pre Rev. 19.3) ACCESSING THE FILE CONCURRENTLY WITH MIDASPLUS. IT IS STRONGLY RECOMMENDED THAT YOU DO NOT USE UNSHARED MIDAS WITH MIDASPLUS.

R-Mode Library

Midasplus now builds an R-mode library interlude with the same name as the former MIDAS R-mode library, KIDALE.

MPLUSLB/VKDALB Synonym Library Names

Since many applications that exist for MIDAS/Midasplus use the shared library name VKDALB (pre Rev. 19.3), Midasplus now builds it's library with two names MPLUSLB and VKDALB.

4 OUTSTANDING PROBLEMS

SPAR 3001371

MPACK loops forever on a file that has all records deleted. This problem is being investigated.

SPAR 3000087

Filere does not work with 'EXT'.

SPAR 3000099

Attempt to access a record number in a range close to the allocated number of records in a direct access file may result in a MIDASPLUS end-of-file error.

5 ENVIRONMENT

MIDASPLUS Rev 19.3 Dependencies:

```
BASICV   - 19.1, 19.2, or 19.3
COFOL      19.3
POWERPLUS 19.1, 19.2, 19.3
PRIMOS     19.3 (will not run prior to 19.3)
```

6 BUILD AND INSTALLATION PROCEDURES

The MIDASPLUS installation procedures are standard using the two command files described below:

MIDASPLUS.INSTALL.COMI: Cominput file that places all the MIDASPLUS files in the appropriate system directories. This file is similar to the standard master disk install files.

MIDASPLUS.SHARE.COMI: Cominput file that properly shares and initializes MIDASPLUS on the system. This file is similar to the standard master disk share files.

The initial directory comes with everything built and ready for installation. This version of MIDASPLUS includes all the online, the offline routines and the MIDAS utilities (CREATK, KBUILD, KIDDEL, MDUMP and MPACK) as well as a new utility - SPY (see appendix A). In addition both an F-mode library KIDALB, and a synonym library VKDALB are built for compatibility w/previous MIDAS/Midasplus releases.

If you require to use any of the MIDASPLUS config directives (see below) you should create a file called MPLUS.CONFIG containing the required config directives in SYSTEM. Note that you do not need to use config directives, they are optional.

It is recommended that MIDASPLUS.SHARE.COMI be placed in the C_PRMO system initialization file so that the library is installed at each system cold-start.

7 CONFIGURATION PARAMETERS

MIDASPLUS has available configuration parameters that may be set upon system initialization. MIDASPLUS is initialized by the IMIDASPLUS command contained in the MIDASPLUS.SHARE.COMI cominput file. The IMIDASPLUS command will set some configuration parameters as directed in a config file, if specified, and makes the appropriate initialization call to the MIDASPLUS library. Below is a description of the configuration file and how to utilize it.

IMIDASPLUS allows a configuration file treename to be given on the command line, or will optionally display a list of the available config parameters if '-HELP' is supplied on the command line. If a treename is not specified, IMIDASPLUS will assume that the configuration file to be utilized is file MPLUS.CONFIG in the current directory. If the MPLUS.CONFIG file does not exist, then IMIDASPLUS will assume that all system configuration defaults are to be assumed and continue the initialization.

The config file may contain a number of directives that specify MIDASPLUS configuration parameters. These parameters are contained in standard text, one-per-line, within the file. If an error is encountered with a particular directive, IMIDASPLUS will print an error message, assume the default, and continue the initialization.

The available configuration directives are described in the paragraphs that follow. Any parameters not specified by a directive will assume its default value.

SEMAPHORE <semaphore-number>

Specifies the PRIMOS semaphore given by <semaphore-number> utilized for user synchronization. The default for <semaphore-number> is -14.

DEBUG [ON | OFF]

Controls MIDASPLUS debug execution and print options for developer debug. The default for this parameter is OFF. This parameter sets debug control on a system wide basis. It may be set on a per user basis by calls to MSGCTL.

FUNITS <max. number of file units>

Specifies the maximum number of file units per user that MIDASPLUS will use for MIDASPLUS subfiles. (This number does not include file units that are used for main MIDASPLUS segment directories.) The default is 128. The value specified should not be less than the maximun number of MIDASPLUS segment directories a user is likely to have open at a time. In most cases, the value should be at least four times that number. This will allow for four subfiles per MIDASPLUS file to be open.

TIMEOUT <seconds>

Specifies the number of seconds that any user will wait for some internal resource (e.g., locks, buffers) before assuming that the system is hung and aborting the current operation. The argument <seconds> should be a non-negative integer expressing the maximum number of seconds to wait for any resource. If <seconds> is zero, then no timeout will occur and the user will wait indefinitely. The default value is 300 seconds or 5 minutes.

PRINT_ERROR [ON | OFF]

Specifies whether MIDASPLUS should print error messages when fatal errors occur. If ON is given, MIDASPLUS will print the MIDAS error code for any fatal type of error encountered. If a PRIMOS system call error was encountered, it will also print the system error message.

Specifying OFF will cause MIDASPLUS to print no error messages. The default for this parameter is ON. This parameter sets the error print control on a system wide basis. It may additionally be controlled on a per user basis by calls to MSGCTL.

BUFFERS <buffer-count>

Specifies the number of internal file buffers that MIDASPLUS is to utilize. The value of <buffer-count> must be between 2 and 64, inclusive. Giving a low value will utilize less working-set, but increase the number of system I/Os and user wait time for a buffer. Giving a large value will decrease the user wait times, but utilize more working-set. The default value for <buffer-count> is 64. It is not recommended that this parameter be changed except on systems with few MIDASPLUS users and limited memory.

REMOTE_TRANSMIT [ON | OFF]

When set ON, allows outgoing MIDASPLUS access to remote files. When set OFF disallows outgoing remote calls. The default for this parameter is ON.

REMOTE_RECEIVE [ON | OFF]

When set ON, allows incoming requests from other network nodes for access to MIDASPLUS files on this system to be processed. When set OFF, incoming requests are denied. The default for this parameter is ON.

REPORT_DUPS [ON | OFF]

When set ON, the reporting of the existence of duplicate entries by returning a value of one in the returned status code (word one of the ARRAY) is enabled. When set OFF, a status code of one is never returned. This facility is useful, since MIDAS is inconsistent in returning the duplicate indicator and many applications which consider a non-zero returned status to be an error may no longer operate in the same fashion with MIDASPLUS. The default value for this parameter is ON. The parameter is setting a system wide control on the returning of duplicate indicators. It may be controlled on a per user basis by calls to MSGCTL.

REPORT_LOCKED [ON | OFF]

When set ON, the reporting of locked records when performing read operations is enabled (i.e. FIND$, NEXT$ operations). The fact that the record which has been read is locked is indicated by the value of Bit 5 in array word 13 being set to 1 (as in MIDAS). This should

not be confused with Bit 1 of array word 10, which is set to 1 to indicate successful operation of a lock record call. When set to OFF (the default) Bit 5 of array word 13 is always set to zero, regardless of the record's locked status. This directive sets the system wide state for locked record reporting, it may be set on a per-user basis by MSGCTL.

SPY_FNAMES [OFF | ON]

When set ON will save the filenames of open MIDAS files for SPY to use when displaying which files have records locks. When set OFF will not save the filenames. Default OFF.

8_CLEANUP

MIDASPLUS utilizes a different cleanup procedure than does MIDAS. Because of the additional user concurrency controls, more cleanup precautions are needed to clear any problem associated with a user or all users.

The command MPLUSCLUP is utilized to perform cleanup for a MIDASPLUS user. This command is separate from the MCLUP command and does not cleanup the user's MIDAS activity. The syntax for the MPLUSCLUP command is:

MPLUSCLUP [-USER <usernumber> | -ALL]

If no arguments are specified, MPLUSCLUP will cleanup only for the current user. Specification of the <usernumber> argument will cause the associated user to be cleaned-up. Specification of the -ALL argument causes all users to be cleaned up within MIDASPLUS. Only the system console or a user who in Midasplus DEBUG mode, may perform a cleanup for other users.

When the cleanup for one particular user is performed, all system resources that the user may have held are released back to the system. This includes not only internal resources, but releasing of any record locks the user owns (contained in the in-memory cache - see differences below) and the closing of any open MIDASPLUS files. Specifying the -ALL argument will cleanup all resources, except the record locks and user file units. It will not close files for the users. Cleaning up for all users may be done while these users are running. MPLUSCLUP will wait for all users to finish with their current operation and then cause them all to pause while it cleans up. MPLUSCLUP will also report any users that may be hung.

Automatic_Cleanup_

At Rev 19.2 a static on-unit was added to do cleanup for a user

automatically if their application is abnormally interrupted by some error condition. For example if the user breaks by hitting CNTRL-P or the BREAK key, then the static on-unit will be invoked and will proceed to cleanup for this user. All files that the user had open will be closed, any locks held released, and internal system resources released.

9 DIFFERENCES BETWEEN MIDAS AND MIDASPLUS

Because of the different internal design of MIDASPLUS there are some operational differences that may appear to a small number of applications.

Timing

Since MIDASPLUS operates substantially faster than MIDAS, application programs may run into problems due to the changed timing of operations. This is likely to show up in multi-user applications where MIDASPLUS will allow far more concurrent activity than MIDAS. This may appear as concurrency errors or in other less obvious ways.(e.g. CAS version 2.0.8 uses a time stamp as a primary key. This does not provide a unique key value if 2 such operations are attempted within the same second).

Record Locking

Record lock information is now cached in memory instead of marking the record on disk. This means that when the user performs a cleanup or does his last close of the file, all the record locks will be released (this is also true of a system crash). Hence, MIDASPLUS assumes that the application uses record locks on a transient basis during the life of the program execution and does not use record locks across multiple program executions or sessions.

Once all the in-memory, cached record lock entries have been utilized (there are 8000 in the current release), the system resorts back to marking the disk records in a similar fashion to the way that MIDAS does. Record locks set in this manner will not be cleaned up and must be cleared with MPACK. Currently there is no easy manner to detect that this type of record locking has happened for a particular file, but MPLUSCLUP will report the fact that on-disk record locking has been performed and that MPACK should be run to unlock records.

Read only files

MIDASPLUS will now support read-only MIDAS files. At the time of the OPENM$ or NTFYM$ call, MIDASPLUS will test whether the user has write permission to the specified file. If the user does not, no operation will be allowed which can modify the contents of the file (assuming the caller did not specify he wanted write permission explicitly in his call) Hence, a MIDAS file may be protected by LUR ACL rights and

MIDASPLUS will work properly on the file. An error status will be returned if the user attempts to perform some call which will attempt to change the file contents.

Maximum number of concurrently open files

MIDASPLUS will support a maximum limit of 1350 unique MIDAS files being open at one time. Any virgin MIDAS file to which the user does not have write permission requires that the file treename be stored in the MIDASPLUS system. MIDASPLUS will support up to 200 of these types of files. MIDASPLUS will utilize up to 128 PRIMOS file units and currently supports 128 users.

Open and close

There were minor differences in the way that MIDAS and MIDASPLUS opened and closed files. (At Primos revs 18.5 and 19.1 MIDAS file opening and closing requirements have been changed and are identical to those for MIDASPLUS.) MIDAS previously did not require that a call to OPENM$ or NTFYM$ be made before accessing a file via the MIDAS on-line routines (i.e. You could open a file with SRCH$$ and then access it directly with an on-line routine such as FIND$). Although this action caused MIDAS to operate inefficiently it would run. MIDASPLUS does require that a call to OPENM$ or NTFYM$ be made before a file can be accessed via the on-line MIDASPLUS routines. Failure to do this will result in MIDASPLUS error code 23 being returned from the on-line routines.

Opening a file for use by BILD$R, PRIBLD or SECBLD has not changed. In these cases the files must be opened through a PRIMOS call and must not use OPENM$ or NTFYM$.

MIDAS will close the file specified in a call to CLOSM$ even if the file is not a MIDAS file (known to MIDAS via a call to OPENM$ or NTFYM$). This is not the case with MIDASPLUS. MIDASPLUS will only close files via calls to CLOSM$ which have been opened as MIDASPLUS files by calls to OPENM$ or NTFYM$. When MIDASPLUS files are opened by SRCH$$ or TSRC$$, the call to NTFYM$ must be made after the open. When MIDASPLUS files are closed by SRCH$$ or TSRC$$, the call to NTFYM$ must be made before the close.

Reporting of the existence of duplicates

MIDASPLUS will correctly report the existence of duplicates by returning a value of one in the status code returned from the on-line routines (word one of ARRAY). MIDAS did this in a very inconsistent fashion. As a result some programs which did not expect to have a status value of one returned may operate in a different way with MIDASPLUS. The returning of the duplicate indicator may be disabled via a MIDASPLUS config parameter.

MIDASPLUS linkage

The linkage area used by MIDASPLUS is stored in segment %6006 from location %40000.

MSGCTL

The on-line MIDASPLUS routine MSGCTL has been enhanced to allow further per user options to be specified. In MIDAS MSGCTL merely allows error message printing to be controlled on a per-user basis. In MIDASPLUS it controls per user control over error printing, debug mode, reporting of duplicate existence, and reporting of locked records on read operations. The calling sequence of MSGCTL is 'CALL MSGCTL(KEY)' where key may be:

key = 0 error printing off.
key = 1 error printing on.
key = 2 debug mode on.
key = 4 debug mode off.
key = '10 duplicate reporting in status word on.
key = '20 duplicate reporting in status word off.
key = '40 locked record reporting on read operations on.
key = '100 locked record reporting on read operations off.

These key values are additive and multiple key values may be used in the same call. Note that OFF takes precedence over ON if conflicting key values are supplied.

Counts of entries added and deleted

In order to increase performance MIDASPLUS does not write to the midas file the updated values of 'entries added' and 'entries deleted' every time an entry is added or deleted. Instead these counters are updated in memory, and written to the disk when the last user closes the file (by a CLOSM$, a NTFYM$ call, MPLUSCLUP, or automatic cleanup by the static on-unit).

Should the last user fail to close the file the counters in the file will not be correctly updated. In such a situation the counters will always overestimate the number of entries in the file by between 1 and 100 entries. The counters will be corrected by running MPACK on the file.

The above must be borne in mind when using CREATK to display the counter values while the file is in use by MIDASPLUS users.

10 NETWORK SUPPORT

MIDASPLUS allows access to remote files existing on another node in the network. In order to access a remote file the following must be done:

The disk on which the remote file exists must be started up on the local system via FAM II.

MIDASPLUS must be installed on BOTH systems.

The MIDASPLUS config directives REMOTE_TRANSMIT, REMOTE_RECEIVE must not disallow the access.

The remote file access capability in MIDASPLUS is built on top of the NPX mechanism. This mechanism is the low-level facility on which FAM II itself is constructed. When a remote file is opened by MIDASPLUS (via an OPENM$ or NTFYM$ call) MIDASPLUS recognizes that the file is remote and takes the appropriate action. Subsequent MIDASPLUS calls accessing the file will be 'bounced' to the remote system and executed by an NPX slave process on that system. The returned results are returned to the caller by the NPX mechanism. Operations on remote files are handled by MIDASPLUS in a transparent fashion to the application and no special action is required.

Some new MIDASPLUS error codes have been defined related to remote file access. They are as follows:

Error code 90 - remote transmit not enabled.
Error code 91 - remote receive not enabled.
Error code 92 - Network error encountered while attempting access.

At PRIMOS revs 18.5 and 19.1 MIDASPLUS has been modified so that MPLUSCLUP will perform a cleanup on remote systems when executed for a user who has remote files open. The remote cleanup is only performed when MPLUSCLUP is executed without -USER or -ALL arguments. Should it be necessary to perform a MPLUSCLUP with these arguments on a system where remote MIDASPLUS access is being performed, the MPLUSCLUP command must be executed on each node involving remote MIDASPLUS access. In the majority of cases the MIDASPLUS static-on-unit will handle cleanup of the NPX slave if an application abnormally terminates.

11 ERROR CODES

Some new error codes have been added to MIDASPLUS and are not in the MIDAS manual. The meaning of these codes is described below.

Error code 28 - Attempt to write to read-only file.
Error code 40 - Fatal internal error within MIDASPLUS.
Error code 41 - Timeout occurred while attempting to get a buffer.
Error code 85 - (ADD19) All index subfiles are full or all data subfiles are full.
Error code 90-99 - Network errors.
Error code *10001 - Error on close.
Error code *10002 - File unit table or shared file table is full.
Error code *10003 - (from OPENM$ only) Not a segment directory.
Error code *10004 - (from OPENM$ only) fatal internal error.
Error code *10007 - (from Utilities and Offline) File in use by MIDASPLUS
Error code -4 - (from GDATA$ only) fatal internal error.
Error code 19 - Full disk condition occurred

during add operation.

Fatal internal errors indicate that an internal error within MIDASPLUS has been detected. These errors are only returned when there is no other appropriate error code defined which may be used to identify the problem. These errors should not occur. If they do please contact home office.

APPENDIX A - SPY

1 RECORD LOCKS DISPLAY

Midasplus holds data record locks by making an entry to an in memory table, if there is room in this table. The locks entered in this table can be displayed:

1. by user number,
2. by file name (user option must be ON in configuration file), or
3. for the whole system,

according to the option chosen by the user. Note that it may be the case that not all the data record locks currently held are entered in this table, for instance if the table was full. The size of the table is presently 8000 entries, if there are ever more than 8000 record locks held then the 8001st lock and those after must be written out to the data record on disk rather then entered into the table. All this means that there may be locks held on disk that are not reflected in the memory table and that will not be displayed under this option. SPY will print a message if there are record locks held on disk. The number of disk locks taken since system initialization and the number currently held on disk can also be displayed with the statistics option.

If the user wants to be able to see the file names of those files that have locks then they must set the FNAME option ON in the MPLUS.CONFIG file. The default for this option is OFF. If ON this option causes the file name to be stored in memory when a file is opened. This costs some in execution time and memory but we expect the cost to be minimal. Only the first sixteen characters of the file name will be saved.

Displaying the record locks held will show the users what locks they and/or others are holding. There should rarely be a need to lock more than a few data records from any given file, at one time.

2 STATISTICS DISPLAY

Statistics are maintained by the system as it is running. These may be useful to a Midasplus user and/or system administrator to monitor the efficency of the configuration of their system. The statistics that may be displayed are:

2.1 Product Information

Product name, revision number, level, and date/time last initialized.

NOTE: On cold start, the system date/time is not set until all the products are shared. Midasplus sets the initialization date using the system date, which at that point is not set. This accounts for the "date not set" clause which may appear in the product info display. Midasplus must be re-shared in order for the initialization date to get set.

2.2 Buffer Management Information

There are from two to sixty four buffers available in the Midasplus buffer pool. These are used strictly for index pages, records are not buffered. This number is configurable within these limits. Each user can have at most two buffers reserved for them at any one time. This is transparent to the user and determined by and according to the particular function that the user is presently executing. When a Midasplus function is invoked one of the first things that happens is an attempt to reserve the number of buffers it will need. If it can not reserve the buffers it waits at that point and does not go on until it can get the buffers.

The buffer allocation mechanism employs a least recently used algorithm. A buffer can be thought of as being referenced and used, referenced and unused, or unreferenced and unused. When a buffer is taken it is marked as referenced and used. When it is released by a user it is "demoted" by one user. When it is released by the <u>last</u> user it is considered referenced but unused. If it is touched again, when the buffer manager is searching for a buffer, then it is marked unreferenced (and is still unused).

When the buffer manager is searching for a buffer for a given index page, it first looks for that page in the buffer pool, buffer pages can be shared. As it checks each buffer if it is not the page needed and the buffer is presently marked unused but referenced, it is marked unreferenced - the buffer is now free for the next time. If the page is not found, then the first free (unreferenced and unused) buffer is used to read the page in. If all the buffers were looked at and none were free then rounds is incremented and the first newly freed buffer is used.

The statistics provided on buffer management will provide

information needed to fine tune the system buffer pool size.

1. Number of requests to 'get' an index block buffer.
2. Number of requests to 'release' an index block buffer.
3. Number of waits for buffers to become free.
4. Percentage of calls requiring wait for free buffer
5. Number of getbuffs that accessed the same block as last getbuff.
6. Percentage of times same buffer as last time was hit.
7. Number of getbuffs that found desired block in buffer pool.
8. Percentage of times desired block was found.
9. Number of getbuffs that caused a Prwf$$ disk read.
10. Percentage of calls which caused disk read.
11. Number of requests to create a new index block.
12. Number of times more than one user used same buffer at same time.
13. Percentage of time multiple users shared buffer.
14. Number of Prwf$$ writes of a buffer block.
15. Number of times a buffer block was demoted.
16. Number of waits for in transition buffers, ie. buffer is being read or written.
17. Number of cycles around buffer pool.
18. Percentage of calls which cycled around buffer pool.

2.3 Subfile to Fileunit Translation Information

(Getunit provides subfile to fileunit translation.)

1. Number of calls to getunit.
2. Number of times subfile unit requested was in getunit cache.
3. Percentage of cache hits.
4. Number of times units were all used up.
5. Percentage of times units were all used up.

2.4 Funtion Calls Information

1. Local Midasplus Calls - Number of calls to each Midasplus function:

   Openm$, Closm$, Find$, Find$$, Next$, Next$$,

   Lock$, Updat$, Delet$, Add1$, Gdata$.
2. Number of outgoing remote calls.
3. Number of incoming remote calls.
4. Number of Midasplus remote errors.

2.5 Process Waits Information

These numbers give users an indication of how often processes must wait to get resources. Since the timeout value is configurable, the system can be tuned according to usage. A heavily used system may need a higher timeout value.

1. Number of calls to Snooze.
2. Number of calls to Awaken.
3. Number of re-waits during snooze.
4. Average number of re-waits during snooze.
5. Number of time checks made during a wait.
6. Average number of timechecks per Snooze.
7. Number of timeouts due to locked resources.

2.6 Record Locks Information

1. Number of record lock calls.
2. Number of record lock attempts where record was already locked.
3. Number of record locks actually written to disk.
4. Percentage of calls resulting in disk locks.
5. Number of records currently locked on disk.

3 CONFIGURATION DISPLAY

Midasplus has both system wide and per user configurable parameters. The system wide configurable parameters are set when the system is built. If there exsists a file called MPLUS.CONFIG in the directory SYSTEM the values for these parameters are set using that file, otherwise a default values are used. The per user configuration can be set using calls to MSGCTL. SPY will display the values of these parameters.

3.1 System Configuration

Debug - ON to print debug messages, OFF not to , default - OFF
Print error - ON to print error messages, OFF not to, default - ON
Report dups - ON to report duplicate key entries, OFF not to, default - ON
Report locked - ON to report record already locked by another ON reads, default - OFF
Remote transmit - ON to allow remote calls out, default - ON
Remote receive - ON to allow remote calls in, default - ON
Buffers - number of buffers (from 2 to 64), default - 64
Semaphore - the semaphore number Midasplus will use, default - -14
Timeout - system wait time, default - 300
Funits - maximum file units Midasplus can use, default - 128
Spy_fnames - ON to save file names when a file is opened, default - OFF

3.2 User Configuration

The default for these user configurable parameters is the system default.

Debug - ON to print debug messages
Print error - ON to print error messages
Return dups - ON to return duplicate key status
Return locked - ON to return record already locked status
ON reads

4 USER INTERFACE

Calling Sequence: SPY

There are no arguments to SPY. All of the input is entered when asked for by the SPY menus. SPY will be in CMDNCL so it can be invoked directly by entering its name.

SPY is a separate offline utility that may be run whether Midasplus is being used or not. The interface is self explanatory and simple to use. It is menu driven where the user makes choices about what exactly should be displayed. There are up to two levels of choices, that is after selecting one of the options off of the first menu of options, there may (depending on the option) appear a second menu of options from which the user can further specify her choice. At each level there is an exit option which will allow the user to stop at any point. At the bottom level SPY will request some further information, such as, user number, or file unit for record locks, or time interval and number of intervals for statistics. Once the information at the bottom level has been displayed as requested, SPY will allow the user to continue at the top level menu again or to stop.

Following are some examples:

```
                    [SPY rev 19.3]

                      MAIN MENU

          ENTER                      IF YOU WANT TO:
 ------------------------------------------------------------------

            1                  Display  DATA  RECORD  LOCKS

            2                  Display  SYSTEM  STATISTICS

            3                  Display  SYSTEM  CONFIGURATION

            0                  to  STOP


Please enter <NUMBER> of the option you choose, or Q[uit] to STOP.
>1

                    RECORD LOCKS MENU

          ENTER                      IF YOU WANT TO:
 ------------------------------------------------------------------

            1                  Display  LOCKS  on  a  FILE

            2                  Display  LOCKS  of  a  USER

            3                  Display  ALL LOCKS

            0                  To  STOP


Please enter <NUMBER> of the option you choose, or Q[uit] to STOP.
>1


Enter the FILENAME:
> MYUFD>SPY>TEST>MP_TEST2
```

DATA RECORD LOCKS ON FILE <DIPART>MYUFD>SPY>TEST>MPTEST2

USER #	LOCKS HELD	USER #	LOCKS HELD	USER #	LOCKS HELD
90	51				

51 LOCKS HELD IN MEMORY.

Please enter F for another FILE, <CR> to CONTINUE or , Q[uit] to STOP
>

NOTE: Remember that filename saving is a configurable option of Midasplus and no filenames are saved if it is OFF.

If the FNAME option is OFF in the system configuration then it is an error to choose option 1, LOCKS ON A FILE, from the record locks menu. An error message will be displayed:

SPY_FNAMES option is OFF for Midasplus. SPY can not display locks by FILENAME. See your system administrator if you wish to have the SPY_FNAMES configuration changed.

If the user chooses option 2, LOCKS BY A USER.

Please enter USER number.
> 90

LOCKS HELD BY USER 90

FILE	LOCKS
<DIPART>MYUFD>SPY>TEST>MPTEST2	51

51 LOCKS HELD IN MEMORY OVER 1 FILES.

Please enter U for another USER, <CR> to CONTINUE or , Q[uit] to STOP
>

If the user chooses option 3, ALL LOCKS then the next display will be.

Display LOCKS by FILE or by USER? Please enter U[ser] or F[ile].
>F

DATA RECORD LOCKS HELD BY MIDASPLUS

FILE	LOCKS
<DIPART>MYUFD>SPY>TEST>MPTEST1	1
<DIPART>MYUFD>SPY>TEST>MPTEST3	51
<DIPART>MYUFD>SPY>TEST>MPTEST2	51
<DIPART>MYUFD>SPY>TEST>MPTEST4	50
<DIPART>MYUFD>SPY>TEST>MPTEST5	51

204 LOCKS HELD IN MEMORY OVER 5 FILES
0 LOCKS HELD ON DISK
204 TOTAL

Please enter <CR> to CONTINUE, Q[uit] to STOP
>

NOTE: Any locks held on disk are not displayed on the screen. Midasplus does not know on which files these disk locks are held.

If the fname option is OFF, the filnames are not displayed but instead,

204 LOCKS HELD IN MEMORY OVER 5 FILES
0 LOCKS HELD ON DISK
204 TOTAL

It the user enters U, to display the locks by user then the display will be:

DATA RECORD LOCKS HELD BY MIDASPLUS

USER #	LOCKS HELD	USER #	LOCKS HELD	USER #	LOCKS HELD
89	1	90	51	91	51
92	50	93	51		

204 LOCKS HELD IN MEMORY OVER 5 FILES
0 LOCKS HELD ON DISK
204 TOTAL

Please enter <CR> to CONTINUE, Q[uit] to STOP
>

When the configurations option •3 • is chosen from the MAIN MENU• the next screen will display:

```
                    CONFIGUPATION   VALUES

                     SYSTEM              USER

-------------------------------------------------------------------------
 DEBUG                ON                  OFF
 PRINT ERROR          ON                  OFF
 REPORT DUPS          ON                  OFF
 REPORT LOCKED        ON                  OFF

 REMOTE TRANSMIT      ON
 REMOTE RECEIVE       ON
 BUFFERS              64
 SEMAPHORE           300
 TIMEOUT             -14
 FILE UNITS          128
 SPY_FNAMES        ON

Please enter <CR> to CONTINUE or Q[uit] to STOP.
 >
```

When the system statistics option, 2, is chosen from the Main Menu, the display will be:

SYSTEM STATSISTICS MENU

ENTER	IF YOU WANT TO:
1	Display PRODUCT INFO
2	Display BUFFER INFO
3	Display FILE UNITS INFO
4	Display FUNCTION CALLS INFO
5	Display PROCESS WAITS INFO
6	Display RECORD LOCKS INFO
7	Display ALL OF THE ABOVE
8	For HELP - brief description of each option
0	To STOP

Please enter <NUMBER> of the option you choose, or 0[uit] to STOP.
> 2

(Note: If statistics options 2-7 are chosen the user is then prompted for the length of the display interval.)

PLEASE ENTER LENGTH OF DISPLAY INTERVAL (in 10ths of a second).
IF < 1, STATISTICS WILL BE DISPLAYED ONCE: 0

STATISTICS WILL BE DISPLAYED ONCE.

Hit RETURN to Continue.

--- OR ---

(if the user inputs a length > 0, he/she will then be prompted for the number of display intervals.)

PLEASE ENTER LENGTH OF DISPLAY INTERVAL (in 10ths of a second).
IF < 1, STATISTICS WILL BE DISPLAYED ONCE: 30

PLEASE ENTER <NUMBER> OF INTERVALS.
IF < 1, STATISTICS WILL BE DISPLAYED CONTINUOUSLY: 5

STATISTICS WILL BE DISPLAYED EVERY 3.0 SECOND(S) FOR 5 INTERVALS.

Hit RETURN to Continue.

At this point, the chosen option will be displayed. Examples of each

are shown here:

OPTION 1)

PRODUCT INFORMATION

PRODUCT NAME:	MIDASPLUS
REV NUMBER:	19.3
LEVEL:	120
DATE LAST INITIALIZED:	date not set

OPTION 2)

BUFFER INFORMATION

NUMBER OF CALLS TO GET INDEX BLOCK BUFFER:	10642
NUMBER OF CALLS TO RELEASE INDEX BLOCK BUFFER:	10208
NUMBER OF WAITS FOR FREE BUFFER:	0
PERCENTAGE OF CALLS REQUIRING WAIT FOR FREE BUFFER:	0
NUMBER OF TIMES SAME BUFFER AS LAST TIME WAS HIT:	102
PERCENTAGE OF TIMES SAME BUFFER AS LAST TIME WAS HIT:	0
NUMBER OF TIMES DESIRED BLOCK WAS FOUND:	10259
PERCENTAGE OF TIMES DESIRED BLOCK WAS FOUND:	96
NUMBER OF CALLS WHICH CAUSED DISK READ:	281
PERCENTAGE OF CALLS WHICH CAUSED DISK READ:	2
NUMBER OF REQUESTS TO CREATE A NEW INDEX BLOCK:	0
NUMBER OF TIMES MULTIPLE USERS SHARED BUFFER:	0
PERCENTAGE OF TIMES MULTIPLE USERS SHARED BUFFER:	0
NUMBER OF DISK WRITES:	10
NUMBER OF TIMES BUFFER WAS DEMOTED:	239
NUMBER OF WAITS FOR BUFFERS IN TRANSITION:	0
NUMBER OF CYCLES AROUND THE BUFFER POOL:	4
PERCENTAGE OF CALLS WHICH CYCLED AROUND BUFFER POOL:	0

OPTION 3)

FILE UNITS INFORMATION

TOTAL NUMBER OF CALLS TO GETUNIT:	20708
NUMBER OF GETUNIT CACHE HITS:	19965
PERCENTAGE OF CACHE HITS:	96
NUMBER OF TIMES UNIT REASSIGNED:	0
PERCENTAGE OF TIMES UNIT REASSIGNED:	0

OPTION 4)

FUNCTION CALLS INFORMATION

NUMBER OF INCOMING REMOTE CALLS:	0
NUMBER OF OUTGOING REMOTE CALLS:	0
NUMBER OF MIDASPLUS REMOTE ERRORS:	0
NUMBER OF LOCAL MIDASPLUS CALLS:	
OPEN%$	8
CLOSY$	191
FIND$	6
FINDE$	0
NEXT%	10186
NEXT%$	0
LOCK$	6
UPDAT$	6
DELET$	5
ADD1$	5
GDATA$	0

OPTION 5)

PROCESS WAITS INFORMATION

TOTAL NUMBER OF CALLS TO SNOOZE:	0
TOTAL NUMBER OF CALLS TO AWAKEN:	5
TOTAL NUMBER OF RE-WAITS DURING SNOOZE:	0
AVERAGE NUMBER OF RE-WAITS DURING SNOOZE:	0
TOTAL NUMBER OF TIMECHECKS:	0
AVERAGE NUMBER OF TIMECHECKS PER SNOOZE:	0
NUMBER OF TIMEOUTS DUE TO LOCKED RESOURCES:	0

OPTION 6)

RECORD LOCKS INFORMATION

```
    TOTAL NUMBER OF RECORD LOCK CALLS:                  10191

    NUMBER OF TIMES RECORD WAS ALREADY LOCKED:          0

    TOTAL NUMBER OF RECORD LOCKS WRITTEN TO DISK:       0

    PERCENTAGE OF CALLS RESULTING IN DISK LOCKS:        0

    CURRENT NUMBER OF RECORDS LOCKED ON DISK:           0


OPTION 7)

PRODUCT INFO:          PRODUCT NAME:                    MIDASPLUS
                       REV NUMBER:                      19.3
                       LEVEL:                           120
                       DATE LAST INITIALIZED:           date not set

FILE UNITS INFO:       TOTAL NUMBER OF CALLS TO GETUNIT:       20708
                       NUMBER OF GETUNIT CACHE HITS:           19965
                       PERCENTAGE OF CACHE HITS:               96
                       NUMBER OF TIMES UNIT REASSIGNED:        0
                       PERCENTAGE OF TIMES UNIT REASSIGNED:    0

PROCESS WAITS INFO:    TOTAL NUMBER OF CALLS TO SNOOZE:                 9
                       TOTAL NUMBER OF CALLS TO AWAKEN:                 5
                       TOTAL NUMBER OF RE-WAITS DURING SNOOZE:          0
                       AVERAGE NUMBER OF RE-WAITS DURING SNOOZE:        0
                       TOTAL NUMBER OF TIMECHECKS:                      0
                       AVERAGE NUMBER OF TIMECHECKS PER SNOOZE:         0
                       NUMBER OF TIMEOUTS DUE TO LOCKED RESOURCES:      0


                    BUFFER   INFORMATION
-------------------------------------------------------------------------
     NUMBER OF CALLS TO GET INDEX BLOCK:                        10642
     NUMBER OF CALLS TO RELEASE INDEX BLOCK:                    10208
     NUMBER OF WAITS FOR FREE BUFFER:                           0
     PERCENTAGE OF CALLS REQUIRING WAIT FOR FREE BUFFER:        0
     NUMBER OF TIMES SAME BUFFER AS LAST TIME WAS HIT:          102
     PERCENTAGE OF TIMES SAME BUFFER AS LAST TIME WAS HIT:      0
     NUMBER OF TIMES DESIRED BLOCK WAS FOUND:                   10259
     PERCENTAGE OF TIMES DESIRED BLOCK WAS FOUND:               96
     NUMBER OF CALLS WHICH CAUSED DISK READ:                    281
     PERCENTAGE OF CALLS WHICH CAUSED DISK READ:                2
     NUMBER OF REQUESTS TO CREATE A NEW INDEX BLOCK:            0
     NUMBER OF TIMES MULTIPLE USERS SHARED BUFFER:              0
     PERCENTAGE OF TIMES MULTIPLE USERS SHARED BUFFER:          0
     NUMBER OF DISK WRITES:                                     10
     NUMBER OF TIMES BUFFER WAS DEMOTED:                        239
     NUMBER OF WAITS FOR BUFFERS IN TRANSITION:                 0
     NUMBER OF CYCLES AROUND THE BUFFER POOL:                   4
```

```
  PERCENTAGE OF CALLS WHICH CYCLED AROUND BUFFER POOL:  0


  FUNCTION CALLS INFO:
        NUMBER OF INCOMING REMOTE CALLS:                0
        NUMBER OF OUTGOING REMOTE CALLS:                0
        NUMBER OF MIDASPLUS REMOTE ERRORS:              0
        LOCAL MIDASPLUS CALLS: OPENM$                   8
                               CLOSM$                   191
                               FIND$                    6
                               FIND*$                   0
                               NEXT$                    10186
                               NEXT$$                   0
                               LOCK$                    6
                               UPDAT*$                  6
                               DELET$                   5
                               ADD1$                    5
                               GDATA$                   0
  RECORD  LOCKS  INFO:
        TOTAL RECORD LOCK CALLS:                                10191
        NUMBER OF TIMES RECORD WAS ALREADY LOCKED:              0
        TOTAL NUMBER OF RECORD LOCKS WRITTEN TO DISK:           0
        PERCENTAGE OF CALLS RESULTING IN DISK LOCKS:            0
        CURRENT NUMBER OF RECORDS LOCKED ON DISK:               0


OPTION 8)

                    DESCRIPTION OF EACH OPTION
-------------------------------------------------------------------------
  PRODUCT INFO:          Product name, rev number, level number,
                         and date/time last initialized.

  BUFFER INFO:           Statistics on index buffer allocation - calls
                         to get or release buffer, waits for free buffers,
                         shared buffers, disk reads and writes, etc.

  FILE UNITS INFO:       Total number of calls to get unit, number and
                         percent of: cache hits, times units re-assigned.

  FUNCTION CALLS INFO:   Statistics on local Midasplus calls
                         (find$,add1$,etc.) as well as remote calls.

  PROCESS WAITS INFO:    Number of snoozes, wakens, and timeouts, total
                         and average re-waits, total and average timechecks.

  RECORD LOCKS INFO:     Total record lock calls, calls for already locked
                         records, total disk locks written, percent of calls
                         causing disk locks and current number of disk locks.
```

5 ERRORS

Only two kind of errors can occur during execution of SPY, internal system errors and user input errors. Internal system errors are fatal. User input errors can nearly always be trapped as only specific input choices are allowed. If the user makes a detectable error when entering a menu option, she will be given two more chances to enter a valid choice and then SPY will give up and stop. If an out of range or otherwise detectable invalid entry is made for user number or file name then the user will be given two more chances before SPY will give up.

6 COMPATIBILITY

SPY is for use with Midasplus rev 19.3 (or greater) and is not at all compatible with Midas. It will be released with Midasplus 19.3 and will be automatically installed when Midasplus and its other utilities are installed.

APPENDIX B - CREATK COUNT OPTION

Below is an example of the CREATK COUNT option.
The first example is for a file in which the index description values a
re
the same as the actual state of the file. The second example shows a f
ile
in which these values are not correct.

```
CREATK
[CREATK rev 19.3]

MINIMUM OPTIONS? Y

FILE NAME? EXAMPLE1
NEW FILE? N

FUNCTION? USAGE

INDEX? 0

ENTRIES INDEXED:          0
ENTRIES INSERTED:       200
ENTRIES DELETED:          0
TOTAL ENTRIES IN FILE:          200

LAST MODIFIED BY MIDAS REV. 19.3

FUNCTION? COUNT
INDEX? 0

ENTRIES INDEXED:          0
ENTRIES INSERTED:       200
ENTRIES DELETED:          0
TOTAL ENTRIES IN FILE:          200

LAST MODIFIED BY MIDAS REV. 19.3


FUNCTION? USAGE

INDEX? 1

ENTRIES INDEXED:          0
ENTRIES INSERTED:       200
ENTRIES DELETED:        100
TOTAL ENTRIES IN FILE:          100

LAST MODIFIED BY MIDAS REV. 19.3

FUNCTION? COUNT
INDEX? 1
```

```
TOTAL ENTRIES IN THE INDEX:        100

FUNCTION? Q



CREATK
[CREATK rev 19.3]

MINIMUM OPTIONS? Y

FILE NAME? EXAMPLE2
NEW FILE? N

FUNCTION? USAGE

INDEX? 0

ENTRIES INDEXED:           0
ENTRIES INSERTED:      10000
ENTRIES DELETED:           0
TOTAL ENTRIES IN FILE:     10000

LAST MODIFIED BY MIDAS REV. 19.3

FUNCTION? COUNT
INDEX? 0

ENTRIES INDEXED:           0
ENTRIES INSERTED:      10001
ENTRIES DELETED:           0
TOTAL ENTRIES IN FILE:     10001

LAST MODIFIED BY MIDAS REV. 19.3

FILE DESCRIPTOR VALUES HAVE BEEN UPDATED

FUNCTION? USAGE

INDEX? 0

ENTRIES INDEXED:           0
ENTRIES INSERTED:      10001
ENTRIES DELETED:           0
TOTAL ENTRIES IN FILE:     10001

LAST MODIFIED BY MIDAS REV. 19.3

FUNCTION? USAGE

INDEX? 1
```

```
ENTRIES INDEXED:           0
ENTRIES INSERTED:       2000
ENTRIES DELETED:         991
TOTAL ENTRIES IN FILE:          1009

LAST MODIFIED BY MIDAS REV. 19.3

FUNCTION? COUNT
INDEX? 1

TOTAL ENTRIES IN THE INDEX:          1000
THE FILE DESCRIPTOR VALUES DISPLAYED BY
CREATK'S USAGE FUNCTION NO LONGER MATCH
THE ACTUAL INDEX COUNT.

FUNCTION? 0
```

Subject : Pascal

Release : 19.3

Date : July 12, 1982

1) New Functionality

A) The compiler has all functionality that was in 19.2.

B) The at-sign is now allowed as an alternate symbol for the carot.

C) A severity 1 warning is now given whenever the parameter to the TRUNC procedure is an integer, not a real.

D) The second argument to the RESET procedure can now be of type string, in addition to what was previously allowed.

E) A severity 2 error message is now given when a type declaration uses a ':=' instead of a '='. The compiler attempts to recover from this error.

F) The data transfer procedures PACK and UNPACK have been added. Please see external documentation for additional information.

G) 500 %INCLUDE files are now allowed.

H) Error recovery has been improved. Among the errors the compiler recovers better from are invalid constants and labels, invalid type declarations, bad parameters in a parameter list for a procedure or a function, and wrong number of arguments to a predefined procedure or function.

2) Polers fixed.

A) The following user visible bugs have been fixed:

40663 B Mutliply defined case labels are now detected at compile time.

52330 B A severity 1 warning is now given whenever the parameter to the TRUNC procedure is an integer, not a real.

52332 P Invalid type for parameter to procedure or function now detected with clear error message.

58241 B Duplicate of 40663.

60640 F A severity 2 error message is now given when a type declaration uses a ":=" instead of a "=". The compiler attempts to recover from this error.

B) The following internal bugs have been fixed:

99999 C The at-sign is now allowed as an alternate symbol for the carot, as specified by the Pascal standard.

99999 C An extra left parentheses in the parameter list for a procedure or function is detected and an error message is given.

99999 C An extra right parentheses in the parameter list for a procedure or function is detected and an error message is given.

99999 C An error message is given when a REAL constant doesn't have a numeric exponent following the "E" and the optional sign.

99999 C An error message is given when a LONGREAL constant doesn't have a numeric exponent following the "E" and the optional sign.

99999 C An error message is now given when an invalid relational operator is used in an expression.

99999 C In specifying a field width, if the fractional digits part is used when the expression is neither of type REAL nor LONGREAL, an error message is given.

I00397 B The second argument to the RESET procedure can now be of type string.

3) Outstanding Problems.

A) There are some outstanding problems listed on the on-line POLERS data base.

4) Environment.

A) This compiler requires a Rev. 19.3 Primos.

5) Installation and Build Procedures

A) The build requires SPL, FTN, and PMA.

B) The Install and Share are standard.

Subject: PHYSAV, PHYRST

Release: 19.3

Date: November 4, 1983

Rev 19.3

New functionality is included at Rev 19.3 in order to support 300MB fixed head disks. The user visible changes are detailed below.

PHYSAV

If the user states that the disk type to be saved is neither 80MB nor 300MB removeable, then the following extra question will be asked of the user :

300MB FIXED MEDIA, MODEL 4475 (YES/NO)?

These enable him to specify the new disk and thus enable it to be backed up on to tape.

PHYSRT

The following message has been included to indicate the disk type being restored if it is either a new one :

160MB OR 300MB FIXED MEDIA (MODEL 4475)

Subject : PL1G

Release : Rev. 19.3

Date : May 17, 1983

1. New Functionality

   None.

2. Problems Fixed

   A. The following Internal Bugs have been fixed since rev 19.1 :

      I00005 : Problem with nested substring not working. PASS2 was updated to make it work.

      I00013 : Problem with variable defined as builtin function. DECLARE was changed to give correct error message.

      I00015 : Subscripted label in procedure statement should be illegal. Updating "plspass2.tb" to distinguish different labels.

      I00021 : Problem was not reproduced in rev19.2.

      I00022 : User Error: PL1G rule => structure variable can't be aligned.

   B. The following Polers have been fixed since rev 19.1 :

      32862 : Problem not reproducible at rev19.2.

      35897 : Problem with "skip(0) in output file with or without option "ctlasa" Pl1g library routine "p$sout" was changed to solve the problem.

      40854 : The maximum data size PL1G can handle is 32k words (one segment) Added changes in "plspass2.tb" to fix the problem.

      41065 : A new file and doing consecutive writes will lock all records. The problem was not reproduced in rev19.2.

      41645 : All system routine call must use upper case if "LCASE" option used. This is a user error!

      42818 : Accessing DAM files, bomb out with record No. beyond 32768. Not reproducible in rev19.2.

44262 : Substr function didn't work properly if array's index is too large. Delete the "Reduce(K)" in Substr_rule, Pass2 to fix the problem.

45781 : Duplicate problem as 40854. fixed in rev19.3.

45861 : Problem with missing include file, and cpl can't detected. Not reproduced in rev19.2.

45571 : Problem with aggregate assignment. Not reproduced in rev19.2.

45572 : Problem with addition of two bit(16) string result as bit(31). Fixed in rev19.3.

45900 : Problem with aggregate assignment. Duplicate 58304.

46375 : Problem with float dec(14) multiplication. Not reproducible in rev19.2.

47169 : Problem with "dcl a entry(fixed) defined(b)". PASS1, "save_as_locator" was changed to fix this problem.

47200 : Problem with extent of based variable. According to PL1G menu 4 - 5 , this is user's responsibility.

51893 : Problem with Error 230 "Compiler Error" is generated. Not reproduced in rev19.2.

51897 : An error (Invalid char) issued when quoted string is read. Problem not reproduced in rev19.3.

51898 : Rank function didn't work if parameter passed as varying char(1). Updated "pass2.tb" and "pass2" to fix the problem.

53135 : Problem with DPG single step operation. Change "ELSE_FOLLOW" part in Pass1 phase to fix this problem.

56489 : When access a DAM file, if Key not found, TRANSMIT error issued. Updated lib routine "p$read" to issue "Key" condition.

57100 : Problem with built-in function "exp", "log". not reproduced in rev19.2.

57201 : Problem with "access violation" in dbg, "type" function. not reproduced in rev19.2.

57807 : Problem with "skip(0) in output file with or without option "ctlasa". Duplicate 35897.

57808 : Problem with "put string(x) edit (y) (z)". Library routine "P*PUTF" changed to fix unaligned problem.

58085 : Regression in 18.3, got "access-violation". Not reproduced in rev19.1.

58304 : Problem with aggregate assignment for fixed bin and float. Pass2 expand_loops modified to solve problem.

58404 : Problem with incorrect error message. Added new error message 476 into errortext.

59183 : PL1C generate error 230 (Compiler error). The problem was not reproduced in rev19.2

59188 : Problem with variable based its own member. Declare phase, issued error_message in proc "validate".

60494 : Problem with "ENDPAGE" not implemented for TTY output. We might improve it in the future.

60594 : Problem of generating quad instructions. Change CONSTANT "FLOAT_*_MAX_P" TO "FLOAT_*_DBL_P".

60639 : Not reproduced in rev19.2.

61855 : The compiler got access violation after "args-mismatched-procedure". Added error recovery in pass2.

61959 : The compiler gave conflict error message. Added correct error message "476" into Errortext.

S3000678 : Problem with array argument declared as (*)bit(*), PASS2 was updated to make it work.

3. Outstanding Problems

A. There are one outstanding poler that a program compiled with DBG option on will get access violation at run time, but if going through DBG, it works fine. This poler is listed in the on-line POLERS data base.

B. Known internal bugs.

* Passes a pointer which point to the 2nd byte of a word as argument will cause some unaligned problem.

4. Environment

   A. This compiler requires a Rev 19 PRIMOS.

5. Installation and Build Procedures:

   A. The build requires SPL.

   B. The Install and Share are standard.

Subject : PRIME POWER

Release : Rev 19.3

Date : 30 November 1983

New Functionality

POWER PASSWORDS

The POWER system passwords have been removed from the PASSWD file in POWER*. This will not affect the PASSWD SYSTEM command. A utility to convert to the new password format is included in POWERPLUS>TOOLS. This utility will be run by the POWERPLUS install file.

PRIVATE PROCEDURE FILES

POWER now supports PUBLIC and PRIVATE PROCEDURE files. PRIVATE PROCEDURES are stored in the UFD specified during the PROCEDURE CREATE command and are only accessible by the user that created the procedure.

PROC CREATE

The format of the PROC CREATE command has been changed to allow the user to specifiy a public or private procedure:

PROC CREATE PUBLIC or PROC CREATE PRIVATE

PRIVATE will be the default CREATE mode. CREATE PRIVATE will ask for the following information:

```
ENTER DIRECTORY (UFD) NAME :
ENTER DIRECTORY PASSWORD   :
ENTER PROCEDURE NAME       :
ENTER PROCEDURE DESCRIPTION:
```

'*' will be accepted as a valid directory name.

CREATE PUBLIC will create procedure files in the POWRCM UFD.

PROC LIST

PROC LIST or PROC LIST ALL

PROC LIST will list all procedure files created by the current user.

PROC LIST ALL will list all public procedure files.

PROC DELETE

PROC DELETE will allow the user to delete any private procedure file created under his/her LOGIN ID or any public procedure file. Existing procedure files will not have an associated LOGIN ID and can be deleted by any user.

PROCS CREATED BY OTHER COMMANDS

The CREATE, FORM, REPORT, and SCREEN commands allow the user to save responses in a procedure file. When the user requests that responses be saved, the following prompts will be issued:

```
DO YOU WANT TO CREATE A PRIVATE PROC? :
ENTER DIRECTORY(UFD) NAME            :
ENTER DIRECTORY PASSWORD             :
ENTER PROCEDURE NAME                 :
ENTER PROCEDURE DESCRIPTION          :
```

If the user chooses to create a public procedure, the command will omit the prompts for directory name and directory password.

CONVERTING TO THE NEW PROC FORMAT

This enhancement will require a change in the structure of the procedure dictionary file. When REV 19.3 POWER is installed it will be necessary to run a utility to convert POWR##. Existing procedure files will not have associated login ids and will all be marked as public files. The utility to convert to the new PROC format is in POWERPLUS>TOOLS and will be run by the POWERPLUS install command file.

DISPLAY USING SCREEN

COMMAND FORMAT

DISPLAY [<setN>] USING SCREENXX

Where setN is an optional set name or set number and SCREENXX is a valid screen name.

DISPLAY USING SCREEN will display the first record in the set on the specified screen. The bottom line of the screen will display:

ENTER COMMAND ("H" FOR HELP): RECORD: 1

The current record number will always be displayed on the bottom of the screen. The cursor is always positioned after the "ENTER COMMAND ... " prompt.

COMMANDS

H - The HELP command displays the available commands:

END, REDRAW, NEXT, PREVIOUS, TOP, BOTTOM, <N> (ABSOLUTE, +/- (RELATIVE)

END - Clear the screen and return to POWER command level

REDRAW - Redisplay the screen and the current record

NEXT - Display the next record in the set

PREVIOUS - Display the previous record in the set

TOP - Display the first record in the set

BOTTOM - Display the last record in the set

<N> - Display the Nth record in the set

+N or -N - Display the Nth relative record in the set

All commands can be abbreviated to one letter.

DESTROY DELETE OPTION

A -DELETE option has been added to the DESTROY command. DEST <filename> <password> -DELETE will remove the file from the POWER dictionary and also delete the disk file. If the file was created with '*' as the directory name, the user must be attached to the directory containing the file in order to delete it.

TABLE SKIP

When adding records to a file with an interactive add command all or part of a table field may be blank filled by entering '$N' as the next value in the table. '$N' will cause the add to skip to the next field in the record.

LIST SYSTEM COMMAND

The LIST SYS or LIST SYSTEM command will allow the System Administrator to list all POWER files - both public and private.

NULL COMMANDS

POWER will no longer display "INVALID COMMAND" when the user enters a blank command.

DISPLAY FILLER FIELDS

The DESCRIPTORS command will now display FILLER fields.

NUMERIC FIELD NAMES

The CREATE AND CREATE CHANGE commands will now accept the following field type names:

REAL*8	or NUM
REAL	or NUM2
INT	or NUM3
INT*4	or NUM4
DEC	or NUM5
COMP-3	or NUM6

LIST VALIDATION

The LIST VALIDATION command will list all of the validated descriptors and their associated validation files for the currently selected file.

COMMAND SYNTAX:

LIST VAL or LIST VALIDATION

WILL DISPLAY:

DESCRIPTOR	FILE NAME
**********	*********

Where DESCRIPTOR is the name of the descriptor being validated and FILE NAME is the name of the file that contains its validation table. If FILE NAME is blank, the descriptor is validated through a range check.

RANGE_FUNCTION

The RANGE function will display the following values for a specified descriptor in the current or specified set:

LOWEST VALUE

LOWER QUARTILE

MEDIAN

MEAN

UPPER QUARTILE

HIGHEST VALUE

NUMBER OF ENTRIES

The MEDIAN is defined as the value at position (N+1)/2.

The LOWER QUARTILE is the value at position (N+2)/4.

The UPPER QUARTILE is the value at position (3N+2)/4.

The MEAN value is the average of the N set elements.

The number of entries is N, the number of elements in the set

RANGE values can be obtained in two ways - by invoking the RANGE function for a particular descriptor and through REPORTS:

INTERACTIVE_RANGE_FUNCTION

A file must be selected and a set created from that file.

RANGE [set name] <descriptor>

will display the following for the specified descriptor:

RANGE OF <descriptor> IS
****

MIN	LQ	MEDIAN	MEAN	UQ	MAX	NO
---	--	------	----	--	---	--

All values except number of entries, are printed to 4 decimal places.
RANGE can only be used on numeric descriptors.

INVOKING RANGE FROM A REPORT

The REPORT CREATE dialogue will ask "DO YOU WANT AVERAGE OR RANGE" for any numeric descriptor. The user may respond:

YES	(AVERAGE ONLY)
BOTH	(AVERAGE AND RANGE)
RANGE	(RANGE ONLY)
NO	(NO AVERAGE OR RANGE)

If RANGE was selected, the RANGE values will be displayed at the end of the REPORT.

WREN

POWER now supports the WREN terminal. The WREN must be index 9 in the POWER terminal table - therefore TERM** and TERM## should be copied from POWERPLUS>TOOLS. New Functionality Since 19.02

Terminals Supported

POWER now supports terminals which do cursor addressing in YX order as well as those which do addressing in XY order. Support has also been added for terminals which do NOT utilize cursor address codes. Because of this new functionality, changes have been made to the command files which configure the terminal drivers. Therefore the user should note the changes made to the following command files located in POWERPLUS>TOOLS.

PT45.COMI O:L.COMI WREN.COMI FOX.COMI HARD.COMI INFOTON4.COMI PT65.COMI REG100.COMI PERCENT.COMI

Any USER defined command files must also be modified acordingly, and run, before Rev. 19.03 is used.

Problems fixed

The following bugs have been fixed in Rev 19.3 (these are changes since Rev19.0).

PSF 37794 * NOT ACCEPTED IN FILE CREATE IN A PROCEDURE. WHEN * IS USED AS A RESPONSE TO THE QUERY FOR UFD NAME, THE * IS TAKEN AS A COMMENT LINE. (Also 35908, 32653)

PSF 40130 INCORRECT USE OF LINKED FILE DESCRIPTOR WHEN CREATING A REPORT. TRYING TO USE AN ALPHA DESCRIPTOR FROM A LINKED FILE IN A REPORT, POWER WRONGLY ASKS FOR A NUMERIC OUTPUT FORMAT.

PSF 36575 REPORT WRITER DOES NOT PRINT WHOLE DESCRIPTOR WHICH IS TWO LINKS AWAY ALTHOUGH A SIMPLE DISPLAY WORKS OK.

PSF 41488 IF A USER IS CREATING A TEMPLATE, ALL OTHER POWER USERS ARE LOCKED OUT OF FURTHER SELECTS.

PSF 41485 NO HEAD OF FORM GENERATED ON FIRST PAGE OF POWER PRINT OUT (Also 31134)

PSF 34630 In REPORT CREATE, A COMPOUND CONDITION GIVEN IN RESPONSE TO "ENTER CONDITION FOR DISPLAYING" MEMBERS IS NOT FLAGGED AS AN ERROR.

PSF 31542 INCORRECT AND ICONSISTENT DATA IS DISPLAYED IN LINKED FIELDS. DIFFERENT DATA IS LISTED BASED ON THE ORDER OF DISPLAY. WHEN 3 FILES ARE LINKED TOGETHER INCORRECT AND INCONSISTENT DATA IS DISPLAYED.

PSF 35655 BAD DISPLAY WITH LINKED FILES AND LEFT SIDE HEADERS. 6 FILES ARE LINKED TOGETHER. THE MIDDLE ONE IS SELECTED AND DISPLAY OF 12 DESCRIPTORS IS REQUESTED. INCORRECT DATA IS DISPLAYED FOR SOME OF THE FIELDS AND THE IDENTIFIERS ARE MISALIGNED.

PSF 24267 DISPLAY USING FORM ABORTS WITH ILLEGAL ACCESS VIOLATION FORM WAS CREATED WITH MULTIPLE COMPUTED NUMERIC VARIABLES.

PSF 27491 UNABLE TO CALCULATE VARIABLES IN REPORT. PROBLEM SEEMS TO OCCUR WHEN THERE IS A MIXTURE OF SECURITY LEVELS FOR THE VARIABLES IN THE FILE. (Also 32049)

PSF 29583 COMPUTED FIELDS (N1....NX) WILL NOT COMPARE CORRECTLY TO A DECIMAL FILE VALUE.

PSF 27517 A PROCEDURE WITH VARIABLES FOR UFD, PASSWORD, AND FILE NAME FOR A CREATE COMMAND, FAILS TO SUBSTITUTE VARIABLES CORRECTLY.

PSF 34642 WHEN RUNNING A REPORT ON LINKED FILES, BLANK LINES ARE

PLACED IN THE REPORT FOR EVERY OWNER RECORD WHETHER MEMBER RECORDS ARE PRESENT OR NOT.

PSF 45074 CREATE ADD SEQUENCE THAT DEFINES A SYNONYM FOR AN INDEXED FIELD IS ACCEPTED BUT THE NEW FIELD NAME DOES NOT FUNCTION PROPERLY FOR FIND.

PSF 42800 IF PAGE NUMBER AT BOTTOM OPTION IS REQUESTED IN A REPORT, THE PAGE NUMBER IS NOT PRINTED ON THE LAST PAGE OF THE REPORT.

PSF 45070 WITH INDEXED DATE FIELD USING < OR > PARTIAL DATE PRODUCES ERRONEOUS RESULTS DEPENDANT ON THE LAST FULL VALUE USED FOR INDEXED DATE FIND (also 45148)

PSF 37327 REPORT CREATE DOES NOT CORRECTLY PROMPT FOR ALL TABLE DESCRIPTORS AFTER A "NO" ANSWER TO THE PROMPT "DO YOU WANT ALL TABLE ITEMS DISPLAYED".

PSF 42799 THE WORD 'TO' IS NOT OPTIONAL IN THE CHANGE COMMAND

PSF 42797 PROBLEM TRYING TO OVERLAY AUDIT FILE IN POWER*. CREATE SEEMS TO WORK OK BUT THEN FIND ALL ABORTS WITH A 'BEGINNING OF FILE' ERROR ON LOGICAL UNIT 14.

PSF 41912 REPORT CHANGE PRINTS GARBAGE FOR OLD TITLE LINES IF USED TO INCREASE THE NUMBER OF TITLE LINES.

PSF 48172 IT IS NOT POSSIBLE TO GIVE A DESCRIPTOR ONE OF THE FOLLOWING NAMES: E, E1, E2, ... D, D1, D2 ... EE AND DD ARE ALLOWED.

PSF 48171 FIND KEY = 'A' RESULTS IN RECORD BEING SELECTED TWICE. SET CANNOT BE CORRECTLY DUMPED AND ADDED.

PSF 41609 THE SUMS OF FIELDS OF PICTURE 9(8)V99 OR GREATER ARE DISPLAYED WITH A SPURIOUS $.

PSF 41850 IF A KEYWORD OF MORE THAN 20 CHARS IS ENTERED, THE FIRST 20 ARE MADE INTO ONE KEYWORD AND THE REMAINING CHARS BECOME ANOTHER KEYWORD.

PSF 48162 COMBINE OF TWO LARGE SETS (ONE > 3000 ENTRIES) PRODUCES INCORRECT RESULT. PRIMARY KEY IS A DECIMAL PICTURE 9(4).

PSF 44688 COMBINE OF LARGE SETS GETS A COMBINE FORMAT ERROR. COMBINE OF 3 SETS OF 21287, 22959, AND 1712 ENTRIES GETS A ' COMBINE FORMAT ERROR' BUT CREATES A NEW CURRENT SET WHICH MAY OR MAY NOT BE COMPLETE

PSF 44689 SORT ON NAMED SET OF 256 ENTRIES MADE FROM FINDS ON SETS OF 20,0000 ENTRIES GETS A 'RECORD LENGTH INDICATED IS <

1* ERROR MESSAGE. (Also 41910)

PSF 40097 IF A PASSWORD WAS CHANGED FOR A UFD CONTAINING POWER FILES, POWER WILL PRODUCE THE MESSAGE: 'DISK NOT ON SYSTEM' WHEN TRYING TO ACCESS THOSE FILES.

PSF 47247 ERRONEOUS DATE FIELD DISPLAYED WHEN USING REPORT FORMAT.

PSF 45507 REPORT FORMAT MISSING FIRST POSITION OF TITLE THAT BEGINS IN COLUMN 1.

PSF 47221 FIND GT ONLY RETURNS PART OF EXPECTED RECORDS.

PSF 42801 DUMP OR DISPLAY OF DESCRIPTOR > 60 CHARS IS TRUNCATED.

PSF 47479 DISPLAY WITH MORE THAN ONE OPTION DOES NOT USE ALL OPTIONS IF A VERTICAL DISPLAY IS REQUIRED.

PSF 48037 CHANGING A LONG , GREATER THAN 68 CHAR, REPORT TITLE DOES NOT CHANGE THE ENTIRE TITLE

PSF 27497 EQUIVALENT FINDS PRODUCE DIFFERENT RESULTS

PSF 44336 DISPLAY OR PRINT COMMAND DOES NOT DISPLAY ALL TABLE ITEMS

PSF 41852 AN OVERLAID MIDAS FILE WITH BIT STRING AND ASCII DESCRIPTORS AND KEYWORDS FAILS TO PERFORM CORRECTLY ON A FIND KEYWORD THAT FOLLOWS A FIND OF AN ASCII DESCRIPTOR.

PSF 47477 WHEN USER HAS A FULL COMPLEMENT OF SETS POWER FAILS TO FIND ENTRIES AND BECOMES CONFUSED ON NUMBER OF SETS SAVED

(Changes since rev 1*.1):

PSF 29703 FIND <search-expression> AND <search-expression> dooes not work properly with tables.

PSF 41468 Descriptors from linked files do not appear on reports with more than two rows.

PSF 41*94 DUMP of a set gives bad data in a date field from a linked file which is 2 links away from the current file.

PSF 43565 If the field length is defaulted in a CREATE CHANGE, a DESCRIPTOR OUT OF RANGE message is displayed

PSF 44928 FIND with OR on MIDAS primary key not working correctly.

PSF 45141 In a report, incorrect subtotals are displayed for the 9th descriptor with subtotals.

PSF 45887 COMBINE on two disjoint large sets produces a non-empty set.

PSF 45926 POWER temporarily changes the user's LWORD to suppress password echoing. This can cause a forced logout over dial-in lines.

PSF 47925 POWER does not allow a TABLE to start at character position > 5000 even if the record size is sufficient.

PSF 52199 PRINT -AT DS.ONE results in printer DS and FORM ONE.

PSF 27496 Illegal numeric output format not detected during the creation of a new form.

PSF 40135 Unable to use more than one POWER function in an expression. i.e. N1 = SUM(desc1)*AVG(desc2) would result in N1 equaling only SUM(desc1).

PSF 51051 DISPLAY USING REPORTxx causes user terminal to hang if the report contains fields from linked files.

PSF 58301 Page number and column headings are scrolled off the screen during a display.

PSF 60623 Table descriptors are now being validated correctly during a non-screen ADD.

PSF 60775 MIDASPLUS no will no longer give the error message 'UNIT NOT OPEN' when using POWERPLUS with linked files. This also allows OAS users to run correctly after a POWERPLUS session with linked files.

(Changes since rev 19.2):

PSF 43069 During a CHANGE USING SCREENxx, a change made to the first field of the first record will no longer be propagated to all the records in the set.

PSF 45143 The selected security level of a descriptor during a CREATE CHANGE will now REPLACE the current security level instead of being ADDED to it.

PSF 53404 The FORM DELETE command will no longer delete the wrong form, and the FORM LIST command will now give an accurate list of the forms available.

PSF 56891 and 70031 When doing a DISPLAY USING REPORTxx with discriptors from owner.member and selected file will no longer hang.

PSF 57999 The SPLANK option used in a numeric field during a CHANGE USING SCREENxx is now accepted as legal input.

PSF 61870 During an ADD USING SCREEN with a PST-100, when the user gets an input validation error, the screen will no longer scroll up two rows.

PSF 60776 The DISPLAY USING REPORTxx function will now print lines of a report that contain no discriptors (blank lines).

PSF 41974 Performing a DISPLAY USING REPORTxx, where xx is not a valid report number, will no longer cause subsequent displays to print two asterisks instead of the requested report.

PSF 60651 Command file PASSWORD.COMI will now successfully compile the file STPASS.FTN and copy its binary file into POWERPLUS>BINARY.

SPAR 3001124 Reports with descriptors longer than 67 characters will no longer scroll past thier specified length.

SPAR 3001112 The DISPLAY USING FORMxx will now display all the records from an owner file.

Internally reported bugs which are fixed in 19.3:

- IF A FILE TO BE VALIDATED IS LINKED TO TWO VALIDATION FILES USING THE SAME DESCRIPTOR NAMES IN THE VALIDATION FILES AND IN THE FILE TO BE VALIDATED, THE SECOND VALIDATION FAILS.

- VERTICAL DISPLAY DOES NOT USE FORMATS SET DURING HEADING CREATE

- INDIVIDUAL COMMAND PRIORITY DOES NOT ALWAYS FUNCTION IF USER NAME CONTAINS A PERIOD

- CODE ATTACHES TO USER'S HOME UFD PARTITION, NOT PARTITION WITH POWER* WHEN ADDING A TERMINAL TYPE OR CHECKING THE VALALIDITY OF A TERMINAL TYPE FOR A SCREEN FUNCTION.

- ERROR MESSAGES NOT CLEARED DURING CHANGE USING SCREEN FUNCTION.

- FSIO ERROR OCCURS DUPING DISPLAY OF A NUM5 DESCRIPTOR WITH A FORMAT OF MORE THAN 24 CHARACTERS.

- TERM USER does not allow a user number > than 60. The limit has been changed to 128.

- CLOSM$ call changed to CLOSF - file being closed is a non-midas file

- Valioation of descriptors will no longer cause MIDASPLUS errors to occur.

Documentation Additions

PSF 41859 When overlaying a file with keywords, indexes 15-17 should be defined as BIT strings. Appendix B-2 of the POWER GUIDE currently states that keyword should be defined as ASCII.

PSF 45149 It is not possible to add keywords after the text has been initially created. Later additions of keyowrds must be made by editing the keywords as opposed to editing the text.

PSF 51447 The POWER EDITOR can edit a PROCEDURE file of up to 950 lines. It can NOT edit PROCEDURE files of up 1200 lines as stated on page 15-3 of the POWER GUIDE.

CREATE DIALOG The CREATE prompt 'ENTER LENGTH IN CHARACTERS' has been changed to 'ENTER LENGTH IN CHARACTERS OR FIELD TYPE'.

TABLE CREATION When creating a table in an extended options file, the table creation prompts have been changed to:

```
ENTER DESCRIPTOR NAME:  SURFACE
ENTER LENGTH IN CHARACTERS OR TYPE: TABLE
*** ENTER TABLE INFORMATION ***
ENTER TABLE SIZE IN CHARACTERS:  35
ENTER TABLE STARTING POSITION:   25
*** ENTER DESCRIPTOR INFORMATION ***
ENTER DESCRIPTOR POSITION IN TABLE: 1
ENTER MAXIMUM NUMBER OF ENTRIES:  3
ENTER LENGTH IN CHARACTERS OR TYPE:  10
ARE DUPLICATE VALUES PERMITTED:  YES
ENTER SECURITY LEVEL: 0
```

Outstanding Problems

When displaying linked owner discriptors, where there are many owners to one member, incorrect results may occur.

Environment

Rev 19.3 PRIME POWER must be installed with Rev 19.3 PRIMOS, FORTRAN AND MIDAS.

Installation and Build Procedures

It is essential that the install command file supplied in POWERPLUS be used to install 19.3 POWER. This command file will run

the utilities to convert the procedure directory and the password file. ***NOTE: POWER source modules have been divided into individual subroutines. Therefore, the POWERPLUS build and reload files have been changed. POWERPLUSSRC>INFO.RUNO contains a description of the new source files.

SUBJECT: PRIMOS

RELEASE: 19.3.0110

DATE: November 29, 1983

1 *****WARNING*****

Rev 19.3 must be run with at least a rev v disk controller.

2 Increase in maximum number of processes

The total maximum number of processes which can be configured has been increased from 128 to 255. This includes terminal, remote, slave and phantom processes. Assigned lines are not included in this total as they do not use separate processes. They are, however, included in the limit described in the next section.

A maximum of 128 terminals (including assigned lines) can be connected to a system. This total has not changed at rev. 19.3. However, all these 128 processes are now available. Previously, the 128 process limit also included remote, slave and phantom processes.

As a result of the increase in the maximum total number of processes, the default number of reserved file units per user has been reduced from 16 to 8. The default maximum number of file units is still 128. These defaults can be changed using the FILUNT CONFIG parameter.

The maximum total number of file units which can be open simultaneously by all users is 3247 (decimal). This includes 4 extra system units per user (initial, home, current attach points and system unit). Therefore for the

FILUNT reserved_units maximum_units

command, the following defaults hold:

default reserved_units = 8
default maximum_units = 128

and

NUSR * (reserved_units + 4) <= 3247

The following sanity checks have also been put into the coldstart procedure:

1. The NSEG config parameter should guarantee at least 3 segments per configured process. The system outputs a warning message if this is not true.

2. There must be enough physical memory to allow 2 pages per

configured user plus 200 spare working pages in addition to the permanently wired memory (which is output during the coldstart sequence). If there is not enough physical memory, the system outputs an error message and then halts.

3 Modifying Special Device Drivers at 19.3

At Rev 19.3, PRIMOS has been modified to allocate DMA and DMC channels dynamically at coldstart. Prior to rev 19.3 these channels were hard coded in each DIM (Device Interface Module).
To accomplish coldstart allocation of channels, GTCHAN, a routine to allocate channels, has been added and each DIM requiring these channels has been modified to call GTCHAN. If a customer has a 'special' device driver (that is, one NOT supported by PRIMOS) which uses DMA or DMC channels, it will have to be modified to call GTCHAN to get the needed channels. There are a total of 32 DMA (unchanged) and 256 DMC channels for use.
Here is the calling sequence for GTCHAN:

GTCHAN (chan_type, num_chans, dev_addr, chan_id, err_code)

chan_type This is the type of channel being requested, either DMA or DMC. chan_type is 0 for DMA and 1 for DMC. fixed binary (15) - input

num_chans This is the number of channels requested. If it is more than one then chaining is assumed, and the appropriate channels are assigned to this device. The legal values for DMA channels are 1, 2, and 4; and for DMC channels the legal values are any positive integer. (The number of available DMC channels will be an upper limit.) fixed binary (15) - input

dev_addr This is the device address of the device using the channel. It will be needed to record which channels are allocated to which devices if this is determined to be necessary (future expansion). If channels are being obtained for a group of devices then dev_addr will be the lowest device address within the group. fixed binary (15) - input

chan_id This is a return value which contains the actual DMA/DMC channel id (channel number) of the channel allocated to the caller. If DMC channels are requested then this will be the address of the first memory location in the channel. If more than 1 channel is requested, then chan_id is the channel id of the first channel in the 'chain'. fixed binary (15) - output

err_code This is a return value containing the error code. There are 2 types of errors:
1. Unavailable DMA channels (E$UDMA).
2. Unavailable DMC channels (E$UDMC).

(see sections on Error Handling (5.1.5 and 5.2.6)) fixed binary (15) - output

Five areas have to be modified in the driver:

1. Where possible, add the call to GTCHAN after the I/O windows have been secured. It can only be called once per DIM, i.e., GTCHAN can NOT be called again at warmstart.

2. Add the code to handle the E$UDMA/ F$UDMC error returned from GTCHAN.

3. Change the code that sets up the channels to use the correct channels (returned from the GTCHAN routine).

4. Calculate the correct channel control word and output it to the device.

5. Any status from the channel that requires the knowledge of the channel number will have to be modified to use the correct channel.

4 Disk Enhancements at 19.3

Four disk controllers, for a total of sixteen disk drives is now available. The performance of a multi-driver per controller system has also been enhanced.

5 Power Fail Fault Tolerance

When utilized with a modified disk controller, ECO # , the problem of broken disk records due to a power line fluctuation has been eliminated.

6 ICS support changes for 19.3

7 Profile Enhancements at PRIMOS Revision 19.3

Three new features to enhance the security and flexibility of PRIMOS at Revision 19.3 are described here. The first allows the System Administrator to set a minimum length for login passwords and improves password security. The second provides settable WARNING messages in the RDY command. The third provides the REMOVE_REMOTE_ID command and enhancements to the ADD_REMOTE_ID command.

7.1 Minimum Password Length

Currently the SA may require users to have non-null passwords. A logical extension of this feature is allowing the SA to require that passwords have a specific minimum length, something which is quite desirable for added security.

7.1.1 Changes to EDIT_PROFILE

A new command, MINIMUM_PASSWORD_LENGTH (MPWL), has been added to EDIT_PROFILE. Its syntax is:

Minimum_Password_Length <length>

where <length> is an integer between zero and sixteen. A <length> of zero means that there is no minimum password length, and is equivalent to the old "NO_NULL_PASSWORD -OFF". A <length> of one is equivalent to the old "NO_NULL_PASSWORD -ON".

MINIMUM_PASSWORD_LENGTH will still only be able to list users with null passwords, since the encryption algorithm prevents determination of a password's length.

This change is fully compatible with older versions of both PRIMOS and EDIT_PROFILE. Of course, older versions of PRIMOS will not be able to enforce minimum password lengths; older versions will interpret any minimum length as "no null passwords."

7.1.2 Changes to the CHANGE_PASSWORD command

a user attempts to use CHANGE_PASSWORD to set a password which is shorter than the minimum length, the error "Incorrect format for new password" will be returned. If the old password is the same as the new one, the error "New password same as old one" will be returned.

A new option has been added to CHANGE_PASSWORD. The -PROMPT option (no abbreviation) will cause PRIMOS to prompt for the old password in half-duplex mode, just as it does for the new passwords today.

7.1.3 Changes to the CHG$PW gate

The errors returned by CHG$PW now include:

E$BFAP -- One of the passwords had incorrect format.
This includes the new password being
shorter than the minimum length.
E$BPAS -- The old password was incorrect.
E$EXST -- New password same as old one.

E$NVAL -- The user ID could not be found in the SAD. This generally indicates that the SA has deleted the user's entry while the user was logged in, or some serious problem with the SAD.

In addition, if the old password is incorrect, CHG$PW will wait one second before returning to slow down password-cracking programs.

7.2 WARNING Messages

Since Rev. 18 of PRIMOS, commands have had the ability to return any of three statuses: READY, ERROR, and WARNING. Currently users have the ability to set their own READY and ERROR messages with the RDY command. There is no way to know if a command has terminated with a WARNING status; PRIMOS treats it just like READY.

7.2.1 Changes to the RDY command

In order to support warning messages, two new options have been added to the RDY command: -WARNING_LONG (-WL) and -WARNING_BRIEF (-WB). The default warning messages will not change. That is, by default WARNING will be treated the same as READY. In addition, the prompts set by users may now be up to 80 characters long (the old limit was 20). Prompts are stored in dynamically allocated PROCESS-class storage.

7.2.2 The "SYSTEM_STORAGE$" condition

Previously when an error occurred during allocation or deallocation of PROCESS-class storage, a fatal error was always generated. Since such storage is now allocated during ring three initialization, some way of intercepting the problem became necessary. Therefore, when a problem occurs with the dynamic storage allocation mechanism, the "SYSTEM_STORAGE$" condition is signalled. The information structure returns the type of storage and whether it was being allocated or deallocated. Specifically, the information structure looks like:

```
dcl 1 info_struc based,
      2 class char (32) var,
      2 allocate bit (1) aligned,
      2 code fixed bin;
```

class — Storage class which was being allocated/deallocated when the error occurred. Currently this condition is signalled only for PROCESS class storage.

allocate — This bit is set if the error occurred during

allocation, and is reset if the error occurred during deallocation.

code — The specific error which caused the problem.

The default on-unit for this condition generates a fatal error (which causes the user's environment to be reinitialized).

7.3 Maintaining Remote IDs

The two existing commands for maintenance of remote IDs are ADD_REMOTE_ID and LIST_REMOTE_ID. Once the user's list contains 16 system names, it is not possible to remove the ID for one system to add a new system to the list. Also, the password must be supplied on the command line, thus compromising security.

7.3.1 The REMOVE_REMOTE_ID command

The REMOVE_REMOTE_ID (RRID) command allows a user to remove the remote ID for a given system from her remote ID list. Its syntax is:

```
Remove_Remote_ID -ON <system_name>
```

where <system_name> is the node name of the system whose ID is to be deleted. If <system_name> is not in the list, "Not found" will be returned.

7.3.2 Changes to ADD_REMOTE_ID

The ADD_REMOTE_ID command now takes the -PROMPT option. If this option is supplied, ADD_REMOTE_ID will set the terminal to half duplex and prompt for the password rather than forcing it to be entered on the command line. ARID has also been fixed to print an error message if a bad status is returned from FNSID$.

8 Changes to USAGE at PRIMOS Revision 19.3

This document describes changes to the USAGE command at Primos revision 19.3. These changes include improved command handling, improved output format, system initialization time, ability to get statistics for remote systems, and additional memory utilization data.

8.1 The -ON Option

The -ON option has been added. When -ON is used with a remote system name, statistics for that remote system (rather than the local system) are gathered. The remote system specified must be configured for FAM II, but no remote disks need be available.

8.2 System Initialization Time

USAGE now displays the date and time at which the system was cold-started. This information, when used along with the total CPU and I/O times already provided by USAGE, can be very useful in determining overall CPU and I/O utilization.

8.3 New Memory Utilization Statistics

Before Rev. 19.3, USAGE gave statistics only on how many pages of virtual memory a user was using. This was by no means adequate to understand how memory was actually being utilized. Rev. 19.3 USAGE provides the following additional memory utilization statistics:

- Segments available in the system
- Total number of segments in use
- Physical pages available in the system
- Physical pages in use
- Number of wired pages
- Wired pages in use by each user
- Segments in use by each user

8.4 Four Controller Support

USAGE now supports statistics for up to four disk controllers.

8.5 The -DEBUG Option

The -DEBUG option has been added. This option causes a detailed version number to be displayed.

8.6 Abbreviation for -BRIEF

The -BRIEF option may now be abbreviated to -BR. This is the standard abbreviation for -BRIEF.

8.7 Formatting Improvements

USAGE now displays its headings in upper/lower case for improved readability. Column headings have been changed where space permits to make them more user-friendly. Some alignment problems in BRIEF mode have also been corrected.

9 Changes to the FNSID$ Gate

FNSID$ has been changed to accept the K$DELF key. This key causes the entry for the system with the specified name to be deleted from the remote ID list. If no entry for that system is found, F$FNTF is returned.

For security reasons, FNSID$ no longer returns the password when the key is K$SRCH or K$LIST. Instead, it always returns a null string.

9.1 Improved syntax checking for ICS INPQSZ directive

Prior to 19.3, it was possible to specify any value as the length of the ICS input data queue. Primos would adjust invalid lengths to be the next highest legal value. In addition, it was possible to specify a length which was greater than the maximum queue length. At rev 19.3, the specified queue length will be checked to insure that it is a legal value, i.e., one less than a power of 2 (octal 17, 377, for example), and will also be checked to insure that the maximum value (octal 1777) has not been exceeded. If an invalid length has been specified, an error will be reported and Primos cold start will fail.

9.2 Handling of unknown ICS directives

If an unknown ICS directive is specified in the Primos cold start configuration file, it will now be ignored and cold start will continue. Prior to rev 19.3, an unknown directive would have caused cold start to fail.

9.3 Improved verification of ICS2 async configuration

At rev 19.3, Primos has been modified to do more checking of the physical configuration of ICS2 controllers. The goal of these changes was to insure that once a system is cold started, a particular ICS2 line will always have the same logical name (AMLC xx) regardless of whether the system has been warm started or line cards have been inadvertently removed or added. In some cases, if the line configuration is changed, a warm start will fail indicating that the configuration must be corrected.

In addition, the system administrator can now specify what the expected configuration is, so that logical line names can be maintained across subsequent cold starts regardless of certain configuration changes.

9.3.1 Cold start messages and directives

A new cold start directive, ICS CARDS, has been added at rev 19.3 to allow the expected ICS2 configuration to be specified. Use of this directive is optional. If the ICS CARDS directive is not specified for a particular ICS2, no cold start configuration checking is done for that ICS2 only.

The format of this directive is

```
ICS CAPDS dd xxxxxx
```

where dd is the (octal) controller device address
xxxxxx is the (octal) configuration word

The configuration ("xxxxxx") above is specified as the octal equivalent of an unsigned 16-bit binary string representing the slots in which async line cards are expected. A 1 bit means an async line card is expected in the slot; a 0 bit means no async line card is expected in the slot, etc. Bit 1 corresponds to slot 1, bit 2 corresponds to slot 2, etc. For example, suppose async line cards were to be in slots 1, 2, 3, 5, 8, 9, 10, 11, and 12 of ICS2 device 10. The binary configuration word would be 1110100111110000. Converting this to octal would result in 164760, and the directive would be specified as

```
ICS CARDS 10 164760
```

At cold start, some new messages may be printed indicating ICS2 configuration changes. The message

```
Inconsistent cold start configuration for ICS2 device dd:
```

where dd is the (octal) device address, will be followed by one of the following messages:

an async line card has been found where not expected in slot ss

slot ss is empty where an async line card was expected.

where ss is the (decimal) slot number in the ICS2 card cage.

The first message is fatal, i.e. the system will not continue and cold start will fail. Either the expected configuration must be changed, or the indicated line card must be removed for cold start to succeed.

The second message is only a warning, and will occur when the indicated slot is empty or the line card has failed such that Primos believes the slot is empty. Cold start will continue, and logical line names will be allocated as if a line card was present in that slot.

9.3.2 Warm start messages

Some new messages may also be printed indicating ICS2 configuration changes which have occurred since the last cold start. The message

Inconsistent configuration for ICS2 device dd:

where dd is the (octal) device address, will be followed by one of the following messages:

an async line card has been inserted into slot ss.

the async line card in slot ss has been removed or is now inoperable.

the async line card in slot ss is now inoperable.

where ss is the (decimal) slot number in the ICS2 card cage.

The first message is fatal, i.e. the system will not continue and warm start will fail. A system cold start will be necessary.

The other messages are only warnings. Warm start will continue, and logical line names will be allocated as if a line card was present in that slot.

10 POLER_FIXES

10.1 POLER #57310

The command STAT DEV truncated port number to two digits.

Fixed the problem by adding an extra line of code in PRN$ST.FTN, which will read the port number without truncating it.

10.2 REPLY

The REPLY command truncated the last digit on users numbers with 3 digits.

Added some extra code in REPLY$.FTN which will take a users ID number without truncating it.

10.3 POLERs: 41510 (Related polers: 44706, 53162, 43011, 41316)

The interface to icpl_.plp has changed; an unused parameter, status, has been deleted. Cpl.plp now checks the error code com_status after icpl_ returns. Std$cp.plp, init$p.plp, and lister_.plp have been changed to use the new icpl_ calling sequence.

Old interface:

```
dcl icpl_ entry (char(*) var, char(*) var, fixed bin,

fixed bin, fixed bin, char(*) var);

call icpl_ (arg_source, args, status, com_status, src_unit, result);
```

New interface:

```
dcl icpl_ entry (char(*) var, char(*) var, fixed bin,

fixed bin, char(*) var);

call icpl_ (arg_source, args, com_status, src_unit, result);
```

10.4 Changes to Stat for 19.2

In pre-19.2 stat us, if a user was logged through a system to a (another) remote system and not actually logged in (i.e. the login line was LOGIN FOO -ON BAR), running STAT US on the system that the user was logged through on would display a partition name for the user ever though the user was not logged into that system. This has been fixed.

Before fix:

```
User No Line Device  FOO 5 3 <SYSABC> (to BAR )
```

After fix:

```
User No Line Devices FOO 5 3 (to BAR )
```

10.5 Modifying Special Device Drivers at 19.3

At Rev 19.3, PRIMOS has been modified so that if the user assigns an ICS1 synchronous line which is already assigned to that user, the logical connection between the user and the ICS1 is deleted and will have to be reestablished.

10.6 Magtapes

The magtape request completion routine, MTDONE, was rewritten for clarity and to change the algorithm used to compute the number of words transferred. TSMT now returns correct value for "number of words transferred" in all cases.

11 Known Problems

If the CONFIG file directive NSEG is set such that the number of paging records divided by 64 is less than NSEG, then the condition 'PAGING_DEVICE_FULL%' is raised. Since it is possible to get a segment when there is no paging disk space available, phantoms and slaves may not be able to get their ring 3 stacks and therefore either hang (raising this condition) or log out with the message 'User xxx: Phantom tty request' without any indication in any comoutput file.

Subject: PRIMOS2

Release: 19.3

1 New Functionality

None.

2 User Visable Bug Fixes

Model 3 1600/6250 BPI Capable Magtape Drive Operational Note:

The tape density on a model 3 1600bpi/6250bpi capable magtape drive can be set via software or via the density select switch on the front panel of drive, but not both at the same time. When the density is set via a software call to T$MT or an ASSIGN command (which also calls T$MT), the front panel switch is disabled from having any further effect on the tape density. The front panel switch can only be re-enabled by another call to T$MT that enables the switch (see T$MT changes below).

Under PRIMOS2 (DOS), these drives are initialized to enable the front panel density select switch. Under PRIMOS4, density setting is handled at magtape at ASSIGN time. If PRIMOS4 crashes and a tape dump is to be taken, the act of hitting the "MASTER CLEAR" switch causes these drives to revert to their initial state of 1600 bpi with the front panel switch disabled and the dump is therefore taken at this density.

3 Internal Fixes, Not User Visable at 19.3

1.) Deleted unused dummy "stub" routines for old acls and quota's. 2.) Fixed a bug in GPASS with acl ufd's. 3.) Deleted pointers in PMAIN to old acl and quota routines.

4 Outstanding Problems

None.

Subject: PRIMENET

Release: Revision 19.3

Date: November 29, 1983

1 New Functionality

Half Duplex is not supported in this rev.

X.25-1980 support has been added. Fast Select facility is now available. LAPB support has been added. Level 3 timers, including call request, reset request, & clear request timeouts have been implemented. This should alleviate problems (circuit hangs) due to lost packets. Restarts are now sent based on the restart timer. Diagnostic Packets are logged in the network event log file.

Five new commands, CONFIG_NET, START_NET, STOP_NET, MONITOR_RING and FIND_RING_BREAK have been added for rev 19.3 PRIMENET.

Network start-up has been separated from system cold start. The CONFIG directive "NET ON" will no longer enable PRIMENET. To do so, you must issue the new START_NET command from the system console.

The start_net invocation may be added to your C_PRMO file. Be sure it is added BEFORE any remote disks are added. It can be added at the end of C_PRMO, after the date/time are set, so that local users can use the system while the network is initializing.

Here's how to use the new start_net command:

START_NET {confignet_filename} -NODE node_name {-Tracing_Node} {-HELP}

If no filename is specified, the file PRIMENET*>PRIMENET.CONFIG is used.

"-NODE node_name" is required on the command line because of the new global network configuration files. You should specify the local node name (the name of the node you are using).

AN EXAMPLE: if you wish to bring up node ABC on the network:

i) Use CONFIG_NET to build a network configuration file, and call it PRIMENET.CONFIG in PRIMENET*.

ii) Put the following statement in your C_PRMO file

START_NET PRIMENET*>PRIMENET.CONFIG -NODE ABC

It is safe to include both the "NET ON" directive in your CONFIG file and the START_NET command in your C_PRMO file. This will enable you to run different revs of PRIMOS without having to modify these files.

Besides starting NETMAN, the START_NET command attempts to start the ROUTE-THROUGH Server (RT_SERVER). If unsuccessful for some reason, e.g. no phantom slots available, it prints an error message on the console. The administrator can then start RT_SERVER manually by typing PH PRIMENET*>RT.COMI on the console.

The STOP_NET command is used to disable use of the network on the local node. This command effectively just logs out the network server (NETMAN), cleaning up network databases as this occurs.

Usage: STOP_NET {-HELP}

The STOP_NET command also logs out the RT_SERVER.

The Monitor_Ring command will display the statistics for the ring usage from the point of view of this node.

The Find_Ring_Break command will attempt to determine if there is a break in the ring, and its location.

2 User Visible Bug Fixes

Support for FAM-I has been removed.

The maximum number of remote login/remote log-thru users has been increased from 32 to 63.

A new diagnostic code, CD$NSV, has been added to the X$KEYS insert files. NETLINK has been updated to understand this new clearing diagnostic.

New diagnostic codes CD$RTE, CD$LOP, CD$TMO, CD$MEM and CD$RTD have been added to the X$KEYS insert files. Eventually they'll be understood as Route-Through error codes.

SPAR#2001468 : Remote login users may no longer pass off their remote login circuit.

SPAR#2005017 : The acl SLAVES:ALL is no longer needed for PRIMENET*.

SPAR#3000148 : New format TYMNET addresses should no longer cause problems with call connections.

SPAR#3000070 : The CHAP ALL command will no longer lower NETMAN's priority.

LCN 0 is no longer used.

Call collisions are now handled correctly, with only one (not both) of the calls being cleared.

3 Interral Bug Fixes

Netlink now does the "ESCAPE FROM DATA TRANSFER" and "ACTION ON BREAK" parameter functions correctly - it is now possible for a user to disable break transmission in NETLINK, as well as disabling the escape character (usually atsign).

Impure code in VNETLB was cleaned up.

Removed obsolete GFI support: x$agfi & x$ugfi routines were removed from PRIMOS, and their dynts were removed from VNETLB.

4 Outstanding Problems

CONFIG_NET must be run on rev 19.2.2 (or later) PRIMOS, since it utilizes the new S$GRAD gate.

CONFIG_NET may encounter an internal error , abort the configuration session if you give an answer which it does not expect. If you are building a large configuration, you should SAVE periodically.

Subject: PRINT_SYSLOG and PRINT_NETLOG

Release: REV 19.3

Date: November 29,1983

1 Changes at 19.3

o Deleted a '+ 1' in the line setting the value of 'syndrome' in subroutine 'format_check_p9950'.

o Changed references from 4650 to 9950

o Added processing for error codes coming from LYNX PRIMENET.

Subject : RJE Phase 2 Emulator Common Code - RJESRC

Release : 1.5

Date : 09/16/83

1. New Functionality

For a description of new functionality, please refer to IDR4036.

2. Problems Fixed

Emulator Name	RJESRC
Issue No.	1.5
TAR/POLER No.	---*--
Changed by	Suzy
Date Fixed	09/15/83
User Visibility	None.
Fix	The module WK$DIS.PLP has been modified to ensure that on disabling a line the receive sleep timer is reset to a minimum value.

3. Outstanding Problems

There are no known outstanding problems.

4. Environment

This release of the product requires PRIMOS 19.3.

5. Installation and Build Procedures

These are standard.

Subject: ROAM

Date: November 4, 1983

Release: 19.3

1. New Functionality

This is the initial release of ROAM.

2. Problems Fixed

None.

3. Problems Outstanding

1. MAGRST

   a. MAGRST continues restoring a DBMS schema even the user is not a member of .DBMS_ADMIN group.

   b. ROAM bombs out with "system error" while restoring a ROAM file from tape into a ufd where the user does not have sufficient access rights. The user should correct the problem, i.e. have sufficient access (PLUA) to the ufd where the file is to be restored, and re-try MAGRST.

2. COPY_RBF
   ROAM restricts the target filename be unique with respect to the filename on the source machine when copying an active DBMS file to a remote machine.

3. ROAM user file maintenance commands
   All ROAM user file maintenance commands including COPY_RBF, SAVE_RBF, TEST_RBF, CR_RBF, DELETE_RBF, LIST_RBF, and SET_RBF accepts only one command line of 132 characters. ROAM user should be aware of the fact that if he enters a command line of more than 132 characters, only 132 characters will be accepted and no error or warning message is reported.

4. SAVE/ROLLF
   One outstanding bug in MAGLIB will cause an erroneous message reported by both SAVE and ROLLF. When MAGLIB encountered both the end-of-tape and the end-of-file conditions it returns an error code 110 indicating "operation unsuccessful". The SAVE will abort the current SAVE and prompt the user to supply a new tape to start up another SAVE. The ROLLF will report "operation unsuccessful" error and ask the user to supply the correct tape to continue. Since both SAVE and ROLLF treat this consistently no damage will be made and the user can ignore this error.

4. Environment

Rev 19.3 ROAM requires PRIMOS Rev 19.3. ROAM requires the use of shared segments 2200 through 2203, 2217, 2220, 2223, and 2224. ROAM also requires the use of private segments 6007 (words 0 through %47777) and 6011.

5. Installation and Build Procedures

Build -- standard.

Installation and general information to follow.

FILES ON SYSTEM TAPE

ROAM (ufd)

The top level UFD contains the following files: ROAM.INITINSTALL.COMI -- used to install ROAM for the first time, ROAM.INSTALL.COMI & ROAM.INSTALL.CPL -- used to install ROAM after its initial installation, and ROAM.SHARE.COMI -- used to share the ROAM shared segments.

ROAM>CMDNCO

This UFD holds the P-mode interludes to those V-mode segment directories found in SEGRUN* which will become external commands to PRIMOS. These interludes are moved to the top level CMDNCO by ROAM.INSTALL.COMI.

CLUP.SAVE	CN_RBF.SAVE	COPY_RBF.SAVE	DELETE_RBF.SAVE
IROAM.SAVE	LIST_RBF.SAVE	REST_RBF.SAVE	ROSAU.SAVE
SAVE_RBF.SAVE	SET_RBF.SAVE		

ROAM>INFO

This UFD contains the documentation for the current release of ROAM, ROAM.RUNI and ROAM.RUNC.

ROAM>LIB

This UFD contains the ROAM binary object libraries: ROOFFLIB.BIN (the ROAM offline library) and RORUNLIB.BIN (the ROAM runtime library).

ROAM>ROAM*

This UFD contains the following:

3 Files: ROAM.ERR, ROLLB.CPL, TAPE_PH.CPL
2 Segment Directories: ROLLB.SEG, TAPE_PH.SEG
2 Directory: HELP, TOOLS

ROAM.ERR is the ROAM error message file. ROLLB.CPL and ROLLB.SEG are the CPL file and the V-mode segment directory for the ROLLB (Before-Image Recovery) phantom, TAPE_PH.CPL and TAPE_PH.SEG are the CPL file and the V-mode segment directory for the TAPE_PH phantom (used to automatically save the After Image file). The HELP sub-ufd contains ROAM help files.

ROAM>ROAM*>HELP

Help files for ROAM interactive commands. At present, only ROSAU (the help file for the ROAM System Administrator Utility) exists. The subufd ROSAU.UFD contains help files for all ROSAU subcommands.

ROAM>ROAM*>HELP>ROSAU.UFD

Each file in this ufd contains information corresponding to a ROSAU subcommand of the same name, for example, the file ALLOCATE contains information on the ROSAU ALLOCATE subcommand.

ALLOCATE	AUTOSAVE	CLEAR	CONT_AI
CREATE	FIX	HELP	INSTALL
LOCK	MEND	MOVE	QUIT
RECON	RESET	ROLLF	SAVE
STATUS	UNLOCK	WAIT	

The TOOLS sub-ufd contains ROAM tools for system analyst.

ROAM>ROAM*>TOOLS

ROUTL.CPL	ROUTL.SEG	RORUNLIB.MAP	ROOFFLIB.MAP
TRACE_RO.CPL	TRACE_RO.SEG		

The ROUTL.CPL and ROUTL.SEG are the CPL file and V-mode segment directory for the ROUTL tool. The TRACE_RO.CPL and TRACE_RO.SEG are the CPL file and V-mode segment directory for the TRACE_RO tool. Both tools are used to dump or trace runtime ROAM. THE RORUNLIB.MAP and ROOFFLIB.MAP are ROAM runtime and offline shared library maps.

ROAM>SEGRUN*

This UFD contains the V-mode segment directories corresponding to commands in ROAM>CMDNCO.

CLUP.SEG	CN_RBF.SEG	COPY_RBF.SEG	DELETE_RBF.SEG
IROAM.SEG	LIST_RBF.SEG	REST_RBF.SEG	ROSAU.SEG
SAVE_RBF.SEG	SET_RBF.SEG		

ROAM>SYSTEM

This UFD contains the ROAM shared segment run files. The RRxxxx files correspond to the ROAM Runtime shared segments, and the ROxxxx files correspond to the ROAM Offline shared segments.

RO22C3	RO4000	RR2200	RR2201
RR2220	RR4000		

INSTRUCTIONS FOR INITIAL INSTALLATION OF ROAM

1) Restore the ROAM UFD supplied on the system tape.

2) CO ROAM>ROAM.INITINSTALL.COMI
Execute the ROAM initial install program. This program will create the top-level ROAM* ufd on the command device and copy the ROAM system sub-ufds to the appropriate top level system sub-ufds. If desired, the ROAM* ufd may be moved to a partition other than the command device.

3) CO SYSTEM>ROAM.SHARE.COMI
Share ROAM from the system console. Note that when the ROAM initialization program (IROAM) is secged, error messages are printed. IROAM is responsible for restoring ROAM's configuration data, validating the ROAM system files, and initializing ROAM shared memory. Since this is the installation of ROAM, no configuration data nor system files exist, thus these messages may be disregarded.

4) ROSAU
Invoke the ROAM System Administrator Utility's INSTALL subcommand to install ROAM. This entails creating the system files (i.e. the Recovery Table, the Before Image file, and the After Image file) and setting the mode of after imaging. This installation will implicitly initialize ROAM's configuration data. For each system file created The user will be prompted for the UFD in which to place each system file and in the case of the Before and After Image Files the number of pages to initially allocate to each. Creating the After Image file and setting the mode of after imaging (i.e. non-continuous or continuous, manual or automatic saves) is optional depending on whether or not after imaging will be used at your site. The QUIT subcommand returns the user to the PRIMOS command level when complete. The following rules for placement of the system files should be followed.

   A) In order to ensure proper after image recovery, the After Image file should be placed on a different disk partition than the user's database files.

   B) In order to guarantee at least one of either before or after image recovery, in the event of a media failure, the Before Image file should be placed on a different disk partition than the After Image file. In order to increase disk I/O performance, the Before Image file should be placed on a different disk drive than the After Image file.

   C) In order to optimize the reconstruction of the Recovery Table or Before Image file if lost in a media failure, the Recovery Table should be placed on a different disk partition than the Before Image file.

   A rough estimate of the initial size of the Before and After Image files can be obtained by adding the cumulative size of the Before and After Image files for all of your database files.

5) CO SYSTEM>ROAM.SHARE.COMI
   Reshare ROAM again. This time, no error messages should be printed. This command should be included into C_PRM0 in CMDNC0 so that ROAM will be shared any time the system is cold-started.

INSTRUCTIONS FOR RE-INSTALLATION OF ROAM

1) Restore the ROAM UFD supplied on the system tape.

2) CO ROAM>ROAM.INSTALL.COMI
   Execute the ROAM install program.

3) CO SYSTEM>ROAM.SHARE.COMI
   Reshare ROAM.

Subject: Runoff

Release: 19.3

Date: 11/10/83

New Functionality

No new functionality.

Problems Fixed

The following problems were fixed in 19.2.5 and the fixes are included in 19.3.

Unitialized headers and footers are no longer dropped when the page length is shortened.

There was a problem with misalignment on the right margin caused by certain decimal indent commands.

Indent was being set to zero upon Runoff encountering a side margin change or a width change.

Outstanding Problems

None

Environment

This revision of RUNOFF should be build and run on revision 19.0 or later PRIMOS.

Build and Install Procedure

This program may be built and installed by resuming the file RUNOFF>RUNOFF.Build.CPL.

SUBJECT: SEG

RELEASE: Rev19.3

DATE: November 4, 1983

1 New Functionality

- Segment overflow detection for shared subsystems. A new command has been added to check for overflowing of a segment. The command (SO) has three options: OFF, WARN, and FATAL. OFF turns segment overflow checking off; WARN gives a warning message and continues loading; FATAL gives the same warning message and terminates the build. Both WARN and FATAL will return an error prompt, also terminating cominput or cpl execution as well.

2 Problems Fixed

- The problem with loading DAM files has been fixed.
- The symbol table size was increased by about 10%.

Subject: SPOOL

Release: Rev19.3

Date: November 4, 1983

1 Changes Made at Rev 19.3:

2 Internal fixes

1. Changed spooler phantom so that the current time is read in files spooled with the defer option, so that files will be spooled in the correct time.
2. Spool Driver now looks in columns 13 and 15 for form feeds when the -LNUM option is used.
3. Fixed minimum paramters for prop command length and lines.

SUBJECT: SYSCOM

DATE: 11/23/83

REV: 19.3

New Error Code Added error codes of E$BLEF (Bad tape record length and EOF) and E$BLET (Bad tape record length and EOT) for MAGNET.

Subject : V-mode Fortran Library

Release : Rev. 19.3

Date : November 21, 1983

1. New Functionality

None.

2. Problems Fixed

A. Added lon$cn and lon$r as direct entrance calls to the operating system.

B. The following PCLERS have been fixed :

70953 Close could not delete a scratch file if the file was in a subdirectory.

3000782 An overflow on a F-FORMAT, which was a character string variable, caused an 'I/O Error bad FORMAT'.

3. Performance Improvement

The performance has been improved for the input and output of single precision floating point numbers.

4. Outstanding Problems

The outstanding problems with V-mode Fortran Library can be described by customer service, additionally:

With FTN, reading into a REAL*8 variable using a F20.2 format, entering the character string '-0' causes the REAL*8 variable to have a value of 4.4...E-46.

SUBJECT: DBMS/QUERY (VISTA)

RELEASE: Rev 19.3

DATE: July 28, 1983

This document outlines the outstanding problems, environment, and installation procedures for DBMS/QUERY (VISTA), the DBMS Query Report Writer. Included are step-by-step instructions for installation and descriptions of the configuration file and UFD structure necessary for installation.

DBMS/QUERY has new functionality at Rev 19.3. For complete information on the changes, refer to the Rev 19.3 Software Release Document, MRU4304-xxx.

PROBLEMS FIXED

GENERAL

* DBMS/QUERY no longer generates an incorrect "Privacy breach" error when more than 15 areas are used. The limit has been raised to 60 areas. (POLERS # 60201)

* DBMS/QUERY now prints all tokens that are legal (up to the maximum number allowed) when a syntax error occurs.

OUTSTANDING PROBLEMS

GENERAL

* There are no known General problems currently.

REPORT GENERATOR

* DBMS/QUERY incorrectly skips 3 lines when a summary line of SKIP 1 LINE is specified.

* The format compiler aborts when certain complicated arithmetic expressions are used.

ENVIRONMENT

To use Rev 19.3 DBMS/QUERY, it is necessary to install a Rev 19.3 DBMS and a Rev 19.3 ROAM. Also required is a Rev 19.3 operating system (PRIMOS).

INSTALLATION INSTRUCTIONS

To install DBMS/QUERY, take the following actions:

1) If your system uses ACLs, log in under an account that has at least DALURW access on the MFD of the command partition, and the system UFDs CMDNCO and SYSTEM.

2) Attach to the MFD (with the owner password if your system uses passwords) and restore the VISTA tape (this will create the VISTA UFD and its sub-UFDs).

3) If your system uses passwords, modify VISTA>VISTA.INSTALL.COMI to contain the owner passwords for the UFDs CMDNCO and SYSTEM. (If this is your initial installation, make these modifications to VISTA>VISTA.INITINSTALL.COMI, and change the MFD password from XXXXXX to the correct password, if necessary.)

4) Install Rev 19.3 ROAM on the system as per the instructions included with that product.

5) Install Rev 19.3 DBMS on the system as per the instructions included with that product.

6) Modify (if necessary) the configuration file, VISTA>SYSTEM>VISTA.CONFIG (see Addendum 2 of this document for a full description), to fit the system DBMS/QUERY is being installed on. This file will be copied to the SYSTEM UFD; it is the file in SYSTEM which is actually used by DBMS/QUERY.

   NOTE: To reflect a change in the error processing by DBMS/QUERY, the ERROR_TOKENS parameter has been increased from 10 to 48 in the configuration file.

7) Run VISTA.INSTALL.COMI (type "CO VISTA.INSTALL.COMI"). This will copy the necessary UFDs and files from the VISTA UFD to the appropriate system UFDs (see Addendum 1 of this document for a full description of the VISTA UFD). If this is your initial installation, run VISTA.INITINSTALL.COMI instead. This will copy the necessary UFDs and files from the VISTA UFD and create and populate the DBMS/QUERY VISTA* UFD in the MFD. If you wish the VISTA* UFD to reside on a partition other than the command device partition (logical disk 0), it is necessary to modify VISTA.INITINSTALL.COMI, supplying the proper partition name in the appropriate place. Note that the VISTA* UFD <u>should not</u> be an ACLed UFD (for a complete discussion, see Addendum 3).

8) From the system console, share ROAM using SYSTEM>ROAM.SHARE.COMI.

9) From the system console, share DBMS using SYSTEM>DBMS.SHARE.COMI.

10) From the system console, share DBMS/QUERY using the command file SYSTEM>VISTA.SHARE.COMI. This will share the VISTA segments and configure the system based upon the contents of the file SYSTEM>VISTA.CONFIG.

11) DBMS schemas that have been created before Rev 19.3 must be converted to ROAM format before thay may be used by DBMS/QUERY. Refer to the Rev 19.3 DBMS documentation for instructions for doing this.

12) COBOL and FORTRAN subschemas which are going to be accessed by DBMS/QUERY must be deleted and recompiled using the Rev 19.3 COBOL and FORTRAN subschema compilers if the subschemas have not been compiled with post-Rev 18.1 subschema compilers.

13) DBMS/QUERY is now ready for use and may be invoked by typing the command "VISTA" at the terminal. Information about the use of DBMS/QUERY may be obtained from the DBMS/QUERY manuals or by typing the command "HELP" after invoking the subsystem.

ADDENDUM 1: DBMS/QUERY UFD STRUCTURE

The structure of the VISTA UFD (which resides in the MFD and is created when the VISTA tape is restored) is as follows:

VISTA		
	CMDNCO	
		VISTA.SAVE — The command which invokes the Query Report Writer subsystem.
	SYSTEM	
		VI2073, VI2074, VI2075, VI2076 — The shared code and data of DBMS/QUERY; the actual runfile to be shared.
		VISTINIT.SAVE — The program which is run to set DBMS/QUERY's configurable parameters when the system is shared.
		VISTA.CONFIG — The configuration file accessed by VISTINIT.SAVE; contains the configuration parameter values.
	VISTA*	
		VIST_COMMANDS — The file which contains the PRIMOS commands that may be executed from DBMS/QUERY command level by the EXEC PRIMOS command.
		VISTA.ERR — The file which contains the DBMS/QUERY error messages.
		VISTA.LOG — This file will be created by DBMS/QUERY in the VISTA* UFD if a system error occurs; contains information useful in fixing the problem.
		HELP — A UFD which contains DBMS/QUERY's HELP database.
		CATALOG — A UFD structure for the DBMS/QUERY catalogs (created in VISTA* by the command file VISTA.INITINSTALL.COMI).
	INFO	A UFD containing RUNOFF files to help with the installation of DBMS/QUERY.

VISTA.INSTALL.COMI	The COMINPUT file that is run to install DBMS/QUERY on a system where a previous release release of DBMS/QUERY already exists.
VISTA.INITINSTALL.COMI	The COMINPUT file that is run to install DBMS/QUERY on a system for the first time. In addition to the work that VISTA.INSTALL.COMI performs, this file also creates the VISTA* ufd and the user and public catalogs.

Note that neither VISTA.INSTALL.COMI or VISTA.INITINSTALL.COMI (which copy each portion of DBMS/QUERY to the appropriate UFD) deal with the passwords or protection of any of the files or UFDs they copy. Therefore, it is the responsibility of the installer to make sure the new files and UFDs are properly protected and that the COMI files are changed to conform to the passwords present on the system DBMS/QUERY is being installed on. (The suggested protection values for the VISTA* UFDs are all access for owners, none for non-owners: the required protection values for all files copied to the SYSTEM and CMDNCO UFDs are all access for owners, read-only for non-owners.) If ACLs are used on the UFDs SYSTEM and CMDNCO, the following protection should be used:

```
CMDNCO>VISTA.SAVE:     <administrators>:    ALL
                       $REST:               LUR
(Default protection is satisfactory)

SYSTEM>VISTINIT.SAVE:  SYSTEM:              R
                       <administrators>:    ALL
                       $REST:               LUR
(Default protection is satisfactory)

SYSTEM>VISTA.CONFIG:   SYSTEM:              R
                       <administrators>:    ALL
                       $REST:               NONE
```

Use the login name of your system administrator, or the name of the ACL group that is used by system administrators (such as .ADMINISTRATORS) in place of "<administrators>" in the above examples.

ADDENDUM 2: CONFIGURATION FILE FORMAT

The DBMS/QUERY configuration file, SYSTEM>CONFIG.VISTA, consists of 20 lines plus a 4 line header. Each line must be exactly as described in these instructions or DBMS/QUERY cannot be expected to work properly. The file format consists of the information required by DBMS/QUERY followed by an optional comment on each line. A comment begins with "/*" and ends at the end of the line. The information on each line is as follows:

LINES 1-4: Configuration file header. These lines are ignored by VISTINIT.SAVE; they have been added to help document the file. Do not delete these lines; if they are deleted, VISTINIT will still ignore the first 4 lines of the file and will lose necessary information.

LINE 5: The number of characters per line on the TTY DBMS/QUERY is run with. This number should be 1 character less than the actual screen width to avoid unwanted automatic linefeeds. (default = 79) Note: this number should be greater or equal to 71 for optimum performance of DBMS/QUERY.

LINE 6: The number of lines per screen on the TTY DBMS/QUERY is run with. This number should be 1 less than the actual screen length to allow for the scrolling prompt. (default = 23)

LINE 7: The number of characters per logical line on the printer DBMS/QUERY is run with; the number of characters on the line after the printer has inserted its side margins. (default = 132)

LINE 8: The number of lines per logical page on the printer DBMS/QUERY is run with; the number of lines on the page after the printer has inserted its top and bottom margins. (default = 66)

LINE 9: The maximum number of characters per line to be written to a file when the PRINT command FILE option is used. (default = 132)

LINE 10: The maximum number of lines per page to be written to a file when the PRINT command FILE option is used. (default = 66)

LINE 11: With the introduction of ROAM at Rev 19.3, this line is now obsolete and may be left blank. It is still left in the file for compatibility purposes. Formerly, it held the name of the master DBMS UFD (where the schemas are stored).

LINE 12: With the introduction of ROAM at Rev 19.3, this line is now obsolete and may be left blank. It is still left in the file for compatibility purposes. Formerly, it held the owner password of the master DBMS UFD.

LINE 13: The name of the master DBMS/QUERY UFD. (default = *VISTA**)

LINE 14: The owner password of the master DBMS/QUERY UFD. (default = **)

LINE 15: The owner password of the DBMS/QUERY CATALOG UFD (where the procedures, formats and abbrevs are stored). (default = **)

LINE 16: The master UFD of the DBMS/QUERY HELP subsystem files. (default = *VISTA**)

LINE 17: The owner password of the master HELP UFD. (default = **) (Note: If the default master DBMS/QUERY UFD name (VISTA*) is used and the default master DBMS/QUERY HELP Subsystem UFD name (VISTA*) is also used, then the passwords on this line and line 14 must be the same, since the UFDs they apply to are themselves are the same.)

LINE 18: The DBMS/QUERY HELP UFD (the actual data files of the HELP subsystem reside here. (default = *HELP*)

LINE 19: The DBMS/QUERY HELP subsystem UFD owner password. (default = **)

LINE 20: The DBMS/QUERY HELP subsystem topmost level prefix. Since the HELP subsystem prints the actual UFD name where it is currently located, it deletes the topmost (passworded) UFD names and their passwords from the HELP subsystem header. This prefix replaces the deleted portion. (default = *HELP DBMS/QUERY*)

LINE 21: The scrolling default: If *SCROLL ENABLED* is to be the default, set to 1, if *SCROLL DISABLED* is to be the default, set to 0. (default = 1)

LINE 22: The number of virtual records retrieved between printing the virtual record count. (default = 1) Note: If DBMS/QUERY will be used with hard-copy terminals, it is suggested that this number be left as the default. To disable by default the printing of the virtual record count, the System Administrator can create a public DBMS/QUERY "STARTUP" procedure containing the command "DISABLE VIRTUAL RECORD COUNT". This will avoid the constant overwriting of the virtual record count.

LINE 23: The number of "expected" tokens to print after a syntax error. When DBMS/QUERY finds a token it didn't expect it prints what it did expect, up to the maximum number of tokens set by this variable. Users with slow or hard-copy terminals may want to set this to lower than the default value to speed DBMS/QUERY up after syntax errors. (default = 48) Note that DBMS/QUERY now tries to fit as many 'expected' tokens as possible on one line, so that less paper (or lines on a CRT screen) is used.

LINE 24: The name of the public and/or private DBMS/QUERY procedures automatically EXECuted by DBMS/QUERY when entering the subsystem. (default = STARTUP)

ADDENDUM 3: USE OF ACL DIRECTORIES WITH DBMS/QUERY

As of Rev 19.0, the functionality of Access Control List protection of UFDs and files has been added to PRIMOS. This allows the system administrator to specifically state what users are allowed access to files/UFDs and what access they are allowed. This gives each user and the system as a whole much greater control over the amount and type of protection desired. However, DBMS/QUERY requires that users have full access to the VISTA* UFD, but users should not have full access except from within DBMS/QUERY. Therefore, we have chosen to continue using passwords for the present time. VISTA* should not be an ACL directory; if it is, PRIMOS will ignore whatever passwords you choose to configure the UFD with. Note that the user must be attached to a UFD that he/she has write access to for DBMS/QUERY to operate properly; the subsystem creates temporary files.

ADDENDUM 4: PRIMOS COMMANDS FILE FORMAT

The VISTA_COMMANDS file in the DBMS/QUERY master UFD contains the list of PRIMOS commands that may be invoked by the EXEC PRIMOS command. Each line of the file contains the name of a PRIMOS command and optionally the number of characters that it may be abbreviated, with blank lines being ignored. The file may be commented by placing the characters "/*" on a line, and following them with the comment text. The actual format is either just the command, or the command followed by a comma and the minimum abbreviation length. An example is given here:

```
ATTACH,1      /* ATTACH may be abbreviated A, AT, ATT, etc...
COMINPUT,4    /* COMINPUT may be abbreviated COMI, COMIN, etc...
COPY          /* COPY may not be abbreviated.
```

This file is read at DBMS/QUERY initialization time, and the data may not be changed except by reinitializing DBMS/QUERY.

ADDENDUM 5: PRIMOS COMMAND RESTRICTIONS

Certain PRIMOS commands, when executed by the EXEC PRIMOS command can destroy the DBMS/QUERY environment and cause undesirable side effects like returning to PRIMOS command level with DBMS and DBMS/QUERY file units open and leaving an active DBMS transaction running, and these commands must not be included in the VISTA_COMMANDS file. These commands include all commands in CMDNC0 that do not end in a .RUN suffix, and the following internal and .RUN-suffixed commands: BINARY, CLOSE, CPL, DELSEG, INPUT, LISTING, LOGIN, LOGOUT, OPEN, REN, RESTOR, RESUME, RLS, RWLOCK, SAVE, and START.

Information on Rev 19.3 VRPG

DATE: August 25, 1983

TO: VRPG Users

SUBJECT: Information on Rev 19.3 VRPG

REFERENCE: IDM 5040

New Functionality

Modifications and enhancements have been made to include support for ADDROUT files, improvements to the FORMS interface including enhancements to facilitate the CCF conversion effort, support for Command Key Indicators for use with FORMS, and support for Data Structures and Local Data Area. The -XREF option of the compiler now produces a new cross reference listing, which has been enhanced to show more meaningful information and to add information about data structures and subfields. Each of these new features is described in further detail in this info file.

In addition to VRPG modifications, the EMACS editor language mode for RPG/VRPG has been enhanced considerably. For detailed information refer to the EMACS Revision 19.3 info file.

ADDROUT

OVERVIEW OF ADDROUT

IBM System 3 and System 34 RPG-II supports a mode of file processing known as ADDROUT (address output) file processing. The ADDROUT feature provides the function of reading a disk file in an order different from the physical order of the file. This is accomplished by creating a sort template, known as an ADDROUT file, which contains an entry corresponding to records of the disk file. The order of the entries in the ADDROUT file defines a sortec order on the records of the disk file. An RPG program is capable of reading the disk file in the order specified by the ADDROUT file.

One of the benefits of ADDROUT is that it provides the user with the option of maintaining one copy of a file, while allowing multiple ways of viewing the file. This saves on disk space and saves the user from updating several copies of a file. Whenever the disk file is updated, the ADDROUT file must be recreated or modified in order to maintain an ADDROUT file which represents the current condition of the disk file.

PRIME's VRPG supports ADDROUT for Indexed, Direct Access, and Sequential files, and the support is functionally compatible with that available to IBM users. The only known RPG source incompatibility between PRIME and IBM is that IBM'S ADDROUT files must have a record length of exactly 3, whereas PRIME's ADDROUT files require a record length of 6 for direct access and sequential files and a record length equal to the indexed key length for indexed files.

ADDROUT EXAMPLE

Suppose there exists a four record MIDAS indexed file with the keys 01, 02, 03, 04. Each MIDAS record is 80 characters long. A record contains the key in columns 1 - 2 and a left-justified name in colums 4 - 20. The name of the file is 'INPUT'.

```
INPUT

   01 SMITH
   02 JONES
   03 JOHNSON
   04 WHEELER
```

Now suppose an RPG programmer wishes to read this indexed file in order by name. He chooses to use the ADDROUT feature. To do this, he creates the ADDROUT file 'ADDROU'.

ADDROU

```
03
02
01
04
```

'ADDROU' defines the order in which to read the indexed file to obtain the records in alphabetical order by name. Following is the RPG program that the programmer uses to read the indexed file using the ADDROUT feature.

RPG Program

```
F*
F* THIS PROGRAM PERFORMS THE SEQUENTIAL READ OF AN INDEXED
F* FILE USING AN ADDROUT FILE. THE INDEXED FILE SHOULD BE
F* READ IN THE ORDER DEFINED BY THE ADDROUT FILE WHICH IS
F* ALPHABETICAL BY NAME (COLUMNS 4 - 20). AFTER EACH RECORD
F* IS READ FROM THE FILE 'INPUT', THE RECORD IS WRITTEN TO
F* THE FILE 'OUTPUT'. THEREFORE THE FILE OUTPUT SHOULD BE
F* A SEQUENTIAL FILE IN ORDER BY NAME.
F*
FINPUT   IPE  F  80   FOR 2II     1 DISK
FADDROU  IR   F   2    2  2II      EDISK
FOUTPUT  O    F  80  80             DISK
F     ADDROU  INPUT
IINPUT   AA  01
I                                          1  80 DATA
OOUTPUT  D         01
O                          DATA      80
```

The input file is the indexed file 'INPUT' to be read sequentially, once per cycle using the sequential ADDROUT file 'ADDROU' to define the order. The processing will occur until the last record of the ADDROUT file is read. Output will occur once per cycle to the sequential file 'OUTPUT' and will contain the records of the indexed file in the order read. The execution of this program produces the output file:

OUTPUT

```
03 JOHNSON
02 JONES
01 SMITH
04 WHEELER
```

CONTENT OF THE ADDROUT FILE

An ADDROUT file which processes an indexed file is a sequential file composed of records which specify a primary key of the indexed file. The record length of the ADDROUT file should equal the length of the indexed file key.

An ADDROUT file which processes a direct access file is a sequential file composed of records which specify a relative record number within the direct access file. The record length of the ADDROUT file should be 6 bytes. Each entry in the ADDROUT file should be a valid integer between 0 and the number of records allocated in the direct access file. The entry must be right justified and zero or blank filled. The entry may correspond to a missing (blank) record within the direct access file.

An ADDROUT file which processes a sequential data file (compressed or uncompressed) is a sequential file composed of records which specify a relative record number within the data file. Each entry in the ADDROUT file should be a valid integer greater than or equal to one. Record 1 is the first record in the sequential data file. The record length of the ADDROUT file should be 6 bytes. The entry must be right justified and zero or blank filled. Note, accessing an uncompressed file using ADDROUT is much more efficient than accessing a compressed file using ADDROUT.

PROGRAM SPECIFICATION OF ADDROUT FILES

File statement for the ADDROUT file:

	Columns	Column Description	Required Entry	Notes
	6	spec	F	
	7-14	filename	filename	
	15	file type	I	
	16	designation	R	only one allowed per program.
	17	end of file	E or blank	
	18	sequence	blank	
	19	format	F or U	
	20-23	block length	ignored	
*	24-27	record length	n	where n is file record length.
	28	mode	blank	
*	29-30	key length	n	where n is key length.
	31	rec add type	I	this differentiates an addrout file from a record address limits file.
	32	organization	T	
	33-34	overflow	blank	

35-38	key start	blank
39	extension	E
40	device	DISK
53	continuation	blank
54-59	label exit	blank
66	addition	blank
71-72	condition	blank or Ux, where Ux is external indicator U1-U8. External indicator must match external indicator of the input file.

* Record length and key length requirements.

Indexed file
  key length = file key length
  record length = key length

Direct access file
  key length = 6
  record length = 6

Sequential file
  key length = 7
  record length = 6

File stmt for the input file: (Restrictions and requirements)

Columns	Column Description	Required Entry	Notes
15	file type	I or U	
16	designation	R or S	
17	end of file	blank	
28	mode	R	
31	rec add type	I	defines the input file related to the ADDROUT file. One such file must exist.
40-46	device	DISK	
66	file addition	blank	

Extension statement:

Columns	Column Description	Required Entry	Notes
6	description	E	
11-18	from filename	filename	ADDROUT file.
19-26	to filename	filename	input file related to ADDROUT file.
rest of form		blank	

SAMPLE ADDROUT FILES

Indexed File Support

This is an example of an indexed file with NAME as the key and an ADDROUT file that will allow the indexed file to be read in order by department number (DEPT).

Indexed file

	NAME	DEPT	EMP NO.
c	JOHNSON	123	03
o	JONES	124	02
o	SMITH	101	01
o	WHEELER	130	04

Addrout file

o	SMITH
o	JOHNSON
o	JONES
o	WHEELER

Direct File Support

This is an example of a direct access file that uses employee number (EMP NO) as the relative record number. The ADDROUT file is set to read the direct access file in order by department number (DEPT).

Direct access file

	NAME	DEPT	EMP NO.
c	SMITH	101	01
c	JONES	124	02
o	JOHNSON	123	03
o	WHEELER	130	04

Addrout file

o	000001
o	000003
o	000002
o	000004

Sequential File Support

This is an example of a sequential file that is in order by employee number (EMP NO). The ADDROUT file is set to read the sequential file in order by department number (DEPT).

Sequential file

	NAME	DEPT	EMP NO.
o	SMITH	101	01
o	JONES	124	02
o	JOHNSON	123	03
o	WHEELER	130	04

Addrout file

o	000001
o	000003
o	000002
o	000004

RECOMMENDED PROCEDURE TO CREATE ADDROUT FILES

IBM supports a sort utility that will create ADDROUT files. PRIME does not support such a utility.

For an addrout file which processes a Midas direct access or indexed file, the addrout file can be created by copying the data records to a sequential file (MDUMP Midas command), sorting the file on the desired fields (SORT command) and truncating the records within the file so that only the index keys or the 6 character relative record numbers remain (EDIT command).

For an addrout file which processes a compressed or uncompressed sequential file, the addrout file can be created by making a work copy of the file, prepending the relative record numbers on the records of the work file starting with 1 (EDIT command), sorting the work file on the desired fields (SORT command) and truncating all but the relative record numbers from the records of the work file.

Information on Rev 19.3 VRPG

FORMS

FORMS OVERVIEW

The Prime Forms Management System (FORMS) provides the screen display capability for VRPG users. VRPG has always had the capability to interface with FORMS for screen handling by specifying a file with the device type FORMS. The method of specifying the screen input and output within the VRPG program, however, has been limited. For example, VRPG had restricted the FORMS input to occur through DEMAND statements and had restricted the FORMS output to occur through EXCPT statements. This has now been expanded to allow other methods of performing the screen input and output with the FORMS file type.

MODIFICATIONS TO THE FORMS INTERFACE

1. Old requirement: An input file specifying FORMS as the device had to be a Demand file (D in col 16 of File specification) and had to be read using a READ statement.

Change: The input FORMS file can now be defined as a primary or secondary file and will be read during the normal cycle.

2. Old requirement: An output file specifying FORMS as the device had to be written to using an EXCPT statement and an E in column 15 of the Output specification.

Change: The output FORMS file can be defined using Header, Detail, Exception, or Total output.

3. Old requirement: All commands to FORMS had to be directly sent to the FORMS output file by writing a character string such as "##INVOKE xxx", "##CLEAR ALL", or "##RELEASE".

Change: The first screen invocation "##INVOKE xxx" can be automatically done by entering the screen definition name in columns 54 -59 of the File specification of the output FORMS file. If specified, the "##CLEAR ALL" and "##RELEASE" will also automatically occur at the end of execution. If command keys are used, the FORMS command "##FKEYS ON" will also be automatically done in this circumstance. More information on command key indicators is given in a later section.

4. Old requirement: In order to use FORMS in a VRPG program, an input FORMS file and an output FORMS file had to be defined.

Change: The FORMS processing can now also be defined with one file definition by specifying the FORMS file as a combined file (C in column 15 of the File specification). All the functionality that can be accomplished using both the input and output FORMS file can be accomplished using a combined file.

CONVERSION ISSUES FOR CCP USERS

CCP is the IBM System 3 Communications Control Program. This program handles the communications and resource management of the system. CCP introduced to the RPG user a new type of transaction processing and a new interface to terminals. The transaction processing is expanded to allow the execution of an RPG program which only processes one record of data from the terminal before exiting, thus freeing the terminal and the data base for other users. The terminal interface includes a subroutine which is easily callable from an RPG II application program.

PRIME's goal is to make conversion easier from System 3 RPG-II with CCP to Prime's VRPG with FORMS screen handler. RPG programs under CCP can define an input screen file as a primary, secondary or demand file; can define an output file to be processed with header, detail, total or exception output; can define a screen file to have a combined file type. Therefore, all of the previously mentioned FORMS' modifications will aid in the CCP conversion effort.

NEW <u>*OMIT</u> FEATURE FOR CCP CONVERSION

An additional feature has been added to VRPG to support the CCP conversion. Under CCP, the data that is passed and accepted on input and output often includes some screen information such as the operation code or return code, the data length, and the program or terminal name. In most cases, this information is the first 14 positions of the input and output records. PRIME's FORMS package does not expect this information in the data record. A special field, *OMIT, has been added to VRPG to make it easier for the user to make the conversion to a PRIME system. *OMIT may be added to the input/output record description of the FORMS file. This field has the effect of having the compiler disregard, or omit, any fields which are in the specified 'omit' range. All other fields in the record will have their start and/or end positions shifted left the number of positions indicated in the

end-location of the *OMIT field. The user does not have to take these fields out of the record, nor does he have to re-adjust all the start and end locations of the input/output fields. The omitted fields will still be defined, but not on the input/output record. This special use of *OMIT should not effect non-CCP use, nor will it effect the user who still wishes to make the changes manually.

The following formats should be used with *OMIT:

On Input:

col 6	I
col 7-43	blank
col 44-47	From field-location
col 48-51	To field-location
col 52	blank
col 53-58	*OMIT
col 59-70	blank

On Output:

col 6	O
col 7-31	blank
col 32-37	*OMIT
col 38-39	blank
col 40-43	End position in output record
col 44-70	blank

The following rules should be followed when using *OMIT:

1. *OMIT must be the first field in the input/output record.

2. The from-location in the *OMIT input field must equal 1.

3. On input: any field whose START-location in the record is less than or equal to the END-location of *OMIT will be omitted. If any field has an end-location greater than the end-location of *OMIT, a warning message will be issued, but that field will still be omitted.

4. On output: any field whose end-position in the record is less than or equal to the end-position of *OMIT will be omitted.

5. The omitted fields will still be defined, and can be referenced in any calculation statements, but their values will not be defined by their use in the input/output record. It is suggested that these fields be removed from the calculation statements. Since their values are not set during input or output, unexpected results may occur when these fields are used.

6. *OMIT may only be used with records of files associated with the FORMS device.

7. Any record-identification codes associated with an input record using *OMIT must have their positions modified by the user to reflect the 'shifted' position of that character in the record.

8. Any other attempted use of *OMIT will produce compilation errors.

CCP CONVERSION ISSUES

1. File specification.
An RPG CCP program specifies the screen handling file as using SPECIAL device (columns 40-46) and calling the i/o subroutine SUBR92 (columns 54-59). There is also a continuation line for the file statement which specifies an array argument to the subroutine. The definition of the SPECIAL device screen handling is specified by using an input and an output file or by using one combined file. To convert the File specifications to VRPG File specifications, the SPECIAL device is changed to a FORMS device (columns 40-46), the SUBR92 reference is removed, and the continuation lines are removed. If automatic screen invocation and release is desired, then the screen definition name is entered in the output or combined file specification in place of SUBR92 (columns 54-59). Note that VRPG does not use any array argument for the FORMS file.

2. Output to the FORMS file.
An RPG CCP program requires that the first fourteen positions of the output record to the SPECIAL screen file contain the operation code, output length and terminal name. In some cases, positions 15 through 20 contain the requested program name. To convert to a VRPG program, remove these fields from the Output specifications and shift the remaining data fields, or use the *OMIT feature. Reference Prime's FORMS manual (PDR3040) for more information on the operations that are allowed.

3. Input from the FORMS file.
In an RPG CCP program, the first fourteen positions of the input record of the SPECIAL screen file contain the return code, effective input length and terminal name. To convert to a VRPG program, remove these fields from the input specifications and shift the remaining data fields or use the *OMIT feature.

4. Converting a program which uses the transaction processing of only one record of data before exiting.
An RPG CCP program which uses this Force End of File capability may be converted to a VRPG program that continues to process only one record of data by adding a statement to turn on the LR indicator or may be converted to a VRPG program that allows

multiple transactions to occur. One reason for CCP users to have one record transaction processing is to eliminate concurrency problems. With IBM's CCP only one user can have access to a file at one time. By exiting from the program, the file is free to be used by others. On a Prime system, files can be used concurrently and so many of the programs will benefit by a conversion to a multiple transaction environment.

5. Indicators 91 and 92.
In an RPG CCP program, indicators 91 and 92 are reserved for communication operations. Indicator 91 is used for error flagging. If an error occurs during SPECIAL file handling, then indicator 91 is set on. Indicator 92 is used for exception flagging. If an exception occurs during SPECIAL file handling, then indicator 92 is set on. VRPG does not support this use of indicators. It should not be necessary to remove references to these indicators from the program since they will initially be set off and will not signal error conditions from the processing of the FORMS file. The user will be relying on his own checks, FORMS and VRPG for error handling.

SAMPLE PROGRAMS USING FORMS

Old method: (as seen in the VRPG manual, IDP5040)

```
H                 D
FDUMMY   IP  F       80              DISK
FSCRIN   ID  F       80              FORMS
FSCROUT  O   F       80              FORMS
FSFOFL   O   F       80              DISK
IDUMMY   AA  01
T                                              1    1 DU"
ISCRIN   B-  02
I                                              1   80 DATA
C             FRONT      TAG
C                        SETON                        20
C                        EXCPT
C                        SETOF                        20
C                        READ SCRIN                       50
C                        MOVELDATA      ENDFLD  2
C             ENDFLD     COMP '.$'                    80
C   80                   GOTO END
C                        SETON                        30
C                        EXCPT
C                        SETOF                        30
C                        GOTO FRONT
C             END        TAG
C                        SETON                        40
C                        EXCPT
OSCFOUT  F         01
```

```
O                                          15 '##INVOKE SIMPLE'
O         E         40
O                                          11 '##CLEAR ALL'
O         F         40
O                                           9 '##RELEASE'
OSEQFL    E         30
O                           DATA       80
```

New method using an input and output FORMS file:

```
F*  FORMS PROCESSING
F*
F*     THIS PROGRAM RECEIVES INPUT FROM A FORMS SCREEN AND
F*     WRITES THIS DATA TO THE SEQUENTIAL FILE 'SEQFL'.
F*     THE FORMS SCREEN ROUTINE 'SIMPLE' IS INITIALLY
F*     INVOKED TO PAINT THE FIRST SCREEN FOR THE USER.
F*     THE USER ENTERS DATA WHICH IS READ AND THEN WRITTEN
F*     OUT TO 'SEQFL'.  THE SCREEN IS REFRESHED AFTER EVERY
F*     READ.  THE END OF FILE IS SIGNALLED BY THE USER
F*     ENTERING '/*'.  THE SCREEN IS THEN AUTOMATICALLY
F*     CLEARED AND RELEASED.
F*
FSCRIN    IPE F       80              FORMS
FSCROUT   O   F       80              FORMS          SIMPLE
FSEQFL    O   F       80              DISK
ISCRIN    BB  02
I                                               1  80 DATA
OSCROUT   D         02
O*    THIS REFRESHES THE SCREEN FOR THE NEXT INPUT
O                                           7 '##CLEAR'
OSEQFL    D
O                           DATA       80
```

New method using a combined FORMS file and input realignment:

```
F*  FORMS PROCESSING
F*
F*     THIS PROGRAM RECEIVES INPUT FROM A FORMS SCREEN AND
F*     WRITES THIS DATA TO THE SEQUENTIAL FILE 'SEQFL'.
F*     THE FORMS SCREEN ROUTINE 'SIMPLE' IS INITIALLY
F*     INVOKED TO PAINT THE FIRST SCREEN FOR THE USER.
F*     THE USER ENTERS DATA WHICH IS READ AND THEN WRITTEN
F*     OUT TO 'SEQFL'.  THE SCREEN IS REFRESHED AFTER EVERY
F*     READ.  THE END OF FILE IS SIGNALLED BY THE USER
F*     ENTERING '/*'.  THE SCREEN IS THEN AUTOMATICALLY
F*     CLEARED AND RELEASED.
F*
FSCREEN   CFE F       94              FORMS          SIMPLE
FSEQFL    O   F       80              DISK
ISCREEN   FB  02
1*    THE INPUT WILL BE REALIGNED TO BEGIN IN POSITION 1
I                                               1  14 *OMIT
I                                               1   20CODE
```

```
I                                                      3   40LFN
I                                                      5  140NAME
I                                                     15   94 DATA
OSCREEN  D         02
O*    THIS REFRESHES THE SCREEN FOR THE NEXT INPUT
O                                                      7 '##CLEAR'
OSEQFL   D
O                              DATA       80
```

COMMAND KEYS

COMMAND KEY FUNCTIONALITY WITH FORMS

IBM System 34 RPG-II introduced a set of command key indicators for use within RPG. With System 34 RPG, there are two methods of interfacing with the command key indicators. One is through files defined with the device KEYBORD using the SET and KEY commands. The other method is through files which interface with screens (WORKSTN device on System 34). PRIME's VRPG is offering support of the latter method through the FORMS screen handler.

Old method: When a user wanted to have use of function keys, alternatively or in addition to the normal 'send' or 'transmit' key, he had to check manually the last two characters of his FORMS file buffer for specific characters indicating which function key was used. He could then turn on/off indicators to further condition calculations and/or output.

New method: A user can now incorporate in his program IBM System 34 command keys, KA-KN and KP-KY, which correspond to function keys. The programmer no longer needs to check the last two characters of the file for the function key used. The command key indicators can be used to further condition calculations and/or output. When the VRPG runtime library encounters one of the function keys being used to 'transmit' data, the corresponding command key indicator is turned on, and <u>all</u> other command key indicators are turned off. The operation SETOF can also be used to set off command key indicators at any time. To repeat, these new indicators can <u>only</u> be used in columns 9-17 of calc statements, columns 54-59 of SETOF, or in columns 23-31 of output statements. Any other use of them in a VRPG program will result in errors.

Old method: To have the use of function keys, the program had to extend the record length of the FORMS file by two characters. These last two characters contained the information to indicate which function key was used.

New method: If the users chooses to use the command keys in his program, he does not have to define the function key field in his record, nor does he have to define the record length as two characters longer than the data record. Note that the command key indicators, KA-KN and KP-KY, will still function whether or not the record length is extended.

Old method: To use function keys with FORMS, the user must first have written '###FKEYS ON' to the FORMS file. This informs the FORMS package to respond to function key entries.

New method: If any of the command key indicators, KA-KN or KP-KY, are used in an RPG program and a FORMS file is defined with the screen name given on the file specification, then the command '###FKEYS ON' will be sent to FORMS along with the automatic invocation of the screen.

VRPG PROGRAM EXAMPLE USING COMMAND KEY INDICATORS

Here is an example of a VRPG program using function keys, and it's accompanying FORMS fdl description.

```
H*
 * This is a test program used with FORMS on a PT45 terminal.
 * FORMS is invoked, cleared and data is input by using a
 * 'combined' FORMS file, SCREEN.
 * Function keys KA, KB and KC are used to condition
 * calculations and exception output.
 *
FDUMMY   IP  F       80              DISK
FSCREEN  CP  F       80              FORMS          SIMPLE
FOUTPUT  O   F       80              DISK
 *
IDUMMY   AA  01
I                                                1    1 DUM
ISCREEN  BB  02
I                                                1   82 DATA
 *
C             FRONT     TAG
C                       READ SCREEN                      50
C                       MOVELDATA     SDATA  50
C                       MOVELDATA     ENDFLD  2
C             ENDFLD    COMP '  '                        80
C   80                  GOTO END
C   KA
COR KB
COR KC                  EXCPT
C   NKANKBNKC           SETON                        99
C   99                  EXCPT
C                       SETOF                        KAKBKC
C                       SETOF                        99
C                       SETON                        30
C                       EXCPT
C                       SETOF                        30
```

```
C                       GOTO FRONT
C             END       TAG
 *
O*
OSCREEN  E          30
O                                        7 '##CLEAR'
O*
OOUTPUT  E          KA
O                             SDATA      50
O                                        75 '(f1)'
O          E        KB
O                             SDATA      50
C                                        75 '(f2)'
O          E        KC
O                             SDATA      50
O                                        75 '(f3)'
O          E        99
O                             SDATA      50
O                                        75 '(send, other f-key)'
```

FORMS FDL DESCRIPTION

```
SIMPLE   STREAM
         FIELD DATA, LENGTH 80 JUSTIFY LEFT. INPUT-OUTPUT
         END STREAM
SIMPLE   FORMAT
         DEVICE PT45
         FIELD 'SIMPLE TEST PROGRAM ', POSITION (15,2)
         FIELD 'EACH LINE ENTERED IS WRITTEN TO THE FILE OUTPUT',;
           POSITION (8,3)
         FIELD;
           'HIT SEND OR ANY FUNCTION KEY WHEN DONE TYPING A LINE',;
           POSITION (6,4)
         FIELD 'TYPE $$ TO EXIT', POSITION (17,6);
           BLINK REVERSE VIDEO
         FIELD 'DATA :', POSITION (1,9)
DATA     FIELD POSITION (10,9) LENGTH 80, NOPROTECT
         END DEVICE
         END FORMAT
```

DATA STRUCTURES AND LOCAL DATA AREA

VRPG now allows for the use of data structures and local data areas in programs. A data structure can be used to: (1) define the same internal area multiple times using different data formats, (2) subdivide a field so that either the entire field or its subfields can be referenced, or (3) group fields for easier reference. The local data area is a special type of data structure. A local data area is initialized with data from a user segment and is written out to this segment at the end of program execution. This allows data to be accessed by subsequent programs.

To specify a data structure, make the following entries (columns not mentioned must be blank):

Column 6:	I
Columns 7-14:	The name (maximum of 6 characters) of the data structure, which is optional.
Column 18:	U, if this data structure is to be used as a local data area; blank otherwise.
Columns 19-20:	DS, which identifies this as a data structure.
Columns 75-80:	Comments.

When you use a data structure to group fields, fields from non-adjacent locations on the input record can be made to occupy adjacent storage locations internally. The storage area can then be referenced by the data structure name (on output specifications or RLABL only), or by the subfield names. To reference the entire data structure in factor 1, factor 2, or the result field, assign a subfield name to include the entire data structure, and use that subfield name rather than the data structure name.

To specify the subfields of a data structure, make the following entries. These entries must be made on the line below the DS specification (columns not mentioned must be blank):

Column 6:	I
Columns 44-47:	The number of the record position in which the subfield begins relative to the beginning of the data structure.
Columns 48-51:	The number of the record position in which the subfield ends relative to the beginning of the data structure.
Column 52:	The number of digits (0-9) to the right of the decimal position, if the subfield is numeric; blank, if the subfield is alphanumeric.
Columns 53-58:	The subfield name. This may be the same as an input field name or a result field name.
Columns 75-80:	Comments.

When using data structures, the user should remember the following:

1. A data structure is considered an alphanumeric byte string. The structure is initialized to blanks except for that part of the data structure that is initialized with an array, or if it is used as a local data area. If a subfield is a compile-time array, the array data is placed in the data structure after the data structure has been initialized to blanks.

2. You must assure that numeric subfields are initialized with numeric data prior to their use in CHAIN, LOKUP, COMP, or editing operations.

3. You can redefine a subfield in the data structure by specifying the same or part of the same from/to positions for another subfield.

4. The name of an input field or a result field that is being redefined in a data structure must be specified in the data structure; however, it does not have to immediately precede the subfields redefining it.

5. If a field appears as a data structure name or as a data structure subfield name, the physical space reserved for that field is in the data structure regardless of where the field was defined.

6. The from and to positions specified in a data structure for an input field that is being redefined are relative to the beginning of the data structure, not to the positions that the field occupies in the input record.

7. A subfield can have the same length attributes as other fields or subfields. If arrays are specified as subfields, the length specified must equal the amount of storage required to store the entire array.

The following should be considered when using a data structure as a local data area:

1. Data in the local data area is stored in segment 4030. This method of storing the local data area is temporary. Users of this revision should anticipate a change. To initialize the local data area, a *DELSEG 4030* should be used prior to execution.

2. Data in the local data area may be accessed by subsequent programs which also use local data area. The data is available to the user for the length of his process. The size of the local data area in each program should be the same, to insure accurate results.

The following restrictions apply to the use of data structures:

1. The maximum length of a data structure is 9999 characters. However, if the data structure is defined as a local data area, (U in column 18), the maximum length is 256. The length of a subfield in any data structure is always restricted to 256.

2. The length of a data structure is defined in one of two ways:
   (1) If the data structure name is specified as a field in an input record, the length of the data structure must be the same as the length of the input field.
   (2) If the data structure name is not specified as a field in an input record, the length is defined by the highest to position specified for a subfield.

3. Lookahead fields cannot appear as a data structure or a subfield.

4. Packed or binary numeric fields cannot be specified as a subfield.

5. The RPG reserved words UDATE, UDAY, UMONTH, UYEAR, PAGE and PAGE1 through PAGE7 cannot be specified as a data structure name or a subfield.

6. Array-elements and table names cannot be specified as a data structure name or a subfield.

7. The same subfield name cannot be specified in different data structures, nor can a data structure name be specified as part of a data structure (as a data structure name or subfield).

8. The data structure name, when used, may only be referenced as an input field, as the result field of an RLABL, or as an output field.

9. A data structure name is restricted to 6 characters in length.

10. Only 1 local data area may be used per program.

The following is a short program example illustrating some of the features of data structures and local data area.

```
H                                                       T
F*
F*     THIS PROGRAM USES DATA STRUCTURES.  IT TESTS THE USE
F*     OF THE DATA STRUCTURE SUBFIELDS AND LOCAL DATA AREA.
F*     THE INPUT RECORD IS PUT INTO THE DATA STRUCTURE. THE
F*     STRUCTURES ARE OUTPUT USING THE DATA STRUCTURE NAMES.
F*
FINPUT   IP  F      80            DISK
FOUTPUT  O   F     120            PRINTER
E                     AR1        3  4
IINPUT   AA  01
I                                          1  30 F1
IDASTR       DS
I                                          1  30 F1
I                                          1   20SF1
I                                          3  14 AR1
I                                         15  20 SF3
I                                         21  300SF4
ILDASTR      UDS
I                                          1   5 LDA1
I                                          6  200LDA2
C                      MOVE '*LDA*'    LDA1
C                      Z-ADD1          LDA2
OOUTPUT  H    0303 01
O                                       30 'TEST PROGRAM -'
O                                       50 'DATA STRUCTURES'
OOUTPUT  D  2      01
O                                       25 'DATA STRUCTURE:'
O                           DASTR       60
OOUTPUT  D  2      01
O                                       25 'LOCAL DATA AREA:'
O                           LDASTR      60
```

CROSS REFERENCE LISTING

The following is an example of the new cross reference listing, which gives more specific information, particulary about data structures and data structure subfields.

```
Source File: <PART1>CDA>XREF.RPG
Compiled on: 830713  at: 14:21  by: VRPG Rev 19.3
Options: ERRTTY NOOPTIMIZE XREF LCASE E4V NOBANNER NOSEQCHK STATUS

    1 *         *  ************************************************
    2 *         *    CROSS REFERENCE EXAMPLE
    3 *         *
    4 *         *  ************************************************
    5          H                                                T
    6 *        F*
    7 *        F*  XREF.RPG - DATA STRUCTURE TEST
    8 *        F*    THIS PROGRAM TESTS DATA STRUCTURES. IT TESTS THE
    9 *        F*    -XREF OPTION OF THE COMPILER.  THE PROGRAM USES
   10 *        F*    MULTIPLE DATA STRUCTURES.  ONE IS RELATED TO THE
   11 *        F*    INPUT RECORD, THE OTHER IS A LOCAL DATA AREA. VALUES
   12 *        F*    ARE MOVED TO THE SU-FIELDS IN THE 2ND STRUCTURE.
   13 *        F*
   14         FINP     IP  F  80  80            DISK
   15         FOUT     O   F 120 120            PRINTER
   16         E                     AR1        3  4
   17         IINP     AA  01
   18         I                                          1  30 F1
   19         I1              DS
   20         I                                          1  30 F1A
   21         I                                          1   20SF1
   22         I                                          3  14 AR1
   23         I                                         15  20 SF2
   24         I                                         21  300SF4
   25         I                                          1  30 F1B
   26         I                                          1  14 SF5
   27         I                                         15  30 SF6
   28         IDSTR        UDS
   29         I                                          1   4 DSF1
   30         I                                          5   8 DSF2
   31         I                                          9  12 DSF3
   32         I                                          1  12 DST
   33         C                    EXCPT
   34         C                    MOVE *AAAA*    DSF1
   35         C                    MOVE *BBBB*    DSF2
   36         C                    MOVE *CCCC*    DSF3
   37         OOUT     H         1P
   38         O                                       40 *DATA STRUCTURES*
   39         OOUT     F 11      01
   40         O                         F1A           30
   41         OOUT     D 11      01
   42         O                         DST           12
```

NAME OR SYMBOL	LOC	SIZE	DEC POS	ATTRIBUTES
01				INDICATOR REFERENCES: 17 39 41
1P				INDICATOR REFERENCES: 37
AR1	002120	4		ALPHAMERIC EXECUTION-TIME ARRAY (SUBFIELD) REFERENCES: 16 19
DSF1	001670	4		ALPHAMERIC SUBFIELD REFERENCES: 28 34
DSF2	001704	4		ALPHAMERIC SUBFIELD REFERENCES: 28 35
DSF3	001720	4		ALPHAMERIC SUBFIELD REFERENCES: 28 36
DST	001734	12		ALPHAMERIC SUBFIELD REFERENCES: 28 41
DSTR	001254	12		ALPHAMERIC DATA STRUCTURE REFERENCES: 28
F1	001221	30		ALPHAMERIC DATA STRUCTURE REFERENCES: 17 19
F1A	001544	30		ALPHAMERIC SUBFIELD REFERENCES: 19 39
F1B	001624	30		ALPHAMERIC SUBFIELD REFERENCES: 19
INP	002663			FILE REFERENCES: 14 17
OUT	002157			FILE REFERENCES: 15 37 39 41
SF1	001560	2	0	NUMERIC SUBFIELD REFERENCES: 19
SF3	001574	6		ALPHAMERIC SUBFIELD REFERENCES: 19
SF4	001610	10	0	NUMERIC SUBFIELD REFERENCES: 19
SF5	001640	14		ALPHAMERIC SUBFIELD

```
                                    REFERENCES:      19

SF6         001654      16          ALPHAMERIC SUBFIELD
                                    REFERENCES:      19

0000 ERRORS (VRPG - REV 19.3)
```

User Visible Bug Fixes

A number of corrections were made to PRINTER files concerning the areas of skipping and spacing, overflow, and fetch overflow. Many of the changes correspond to the Poler/SPAR corrections listed below. PRINTER files are now more compatible to the IBM RPG-II PRINTER files.

Poler #34360. When entering file assignments within a command input file or a cpl data block, filenames preceded by spaces created 'illegal filename' or 'file not found'. This has been corrected. Runtime will now trim leading blanks from filenames on assignments.

Poler #37177. The compiler did not check for constant array subscripts less than zero or greater than the number of entries in the array. A severity 3 error is now issued if either of these occurs.

Poler #37182. The #VR operation gave incorrect results when factor1 or factor2 was the same data item as the result. This has been corrected.

Poler #40866. In some circumstances, lookahead fields were not being properly set on the first cycle. This has been corrected.

Poler #40867. It was not possible to begin PAGE numbering at 0. This has now been fixed, as well as defining PAGE fields on input and calculation specifications.

Poler #44591. MOVEA with variable array subscripts created severity 3 compiler error. This now works correctly for all types of array subscripts.

Poler #44592. CHAIN with a variable subscript on an array did not work correctly. This has been corrected.

Poler #44593, 57737. When an EXSR statement referenced a subroutine which was not defined, a severity 4 error occurred, which did not refer to the subroutine. This has been changed to a severity 3 error which refers to the subroutine.

Poler #45959. It was not possible to pass an entire array with RLABL. This will now work correctly.

Poler #45960, 52408. Edit codes 2, 4, B, D, K and M did not work correctly for amounts less than 1. This has been corrected.

Information on Rev 19.3 VRPG

Poler #48342. Spacing and skipping problem was corrected. Unwanted extra lines were appearing in printer files. The problem was that a blank 'space after' entry (column 18 of the output specification) was defaulted to a spacing 1 line after printing. This was incorrect. A blank 'space after' entry should be defaulted to 1 only if no entries were made in 'space before', 'space after', 'skip before', and 'skip after' (columns 17-22).

Poler #48343, 52409. Edit words used with an array did not space correctly. This has been corrected.

Poler #52835. Fetch overflow was not being handled properly under some circumstances. The overflow processing was not occurring when the overflow line was reached a second time. This situation occurred when the output was generated by EXCPT statements and the overflow line was reached again while looping within the calculation. There was no condition within VRPG for sensing the overflow again until the overflow indicator was turned off during the RPG cycle (after detail output processing). Changes have been made to correct this.

Poler #52836, 56302. If the end position in an output specification was missing, no error occurred. A severity 3 error is now issued.

Poler #56310, 61735. There was a problem with accessing records in a MIDAS direct files with more than 32767 records. This has been corrected.

Poler #70959. When a filename ended in an "A" in column 14 of the input specification and a sequence entry in column 15-16 began with "N", the "AN" was being confused with the beginning of "AND". "AND" can be entered in columns 14-16 on the second input line of the record description to describe multiple lines of record identifying codes. VRPG was mistakenly reporting an error because of finding the "AND" used on the first record description line for the file. This also occurred for filenames ending in "O" and the sequence entry beginning with "R". This has now been fixed.

Spar #3000706. There were problems with spacing and skipping with multiple or-lines on an output statement. If any one entry in the space and skip columns of an OR line were blank, then the default for that entry was the entry on the previous OR line. However, the default to entries on the previous line should only have occurred if there were no entries in any of the space and skip columns of the OR line. This has now been corrected.

Information on Rev 19.3 VRPG

Spar #3000743. Sequence checking on numeric compile-time arrays was not being handled correctly. Negative entries were producing incorrect sequence checking results. This has been fixed.

Spar #3000756. When "PAGE" fields were being used with output indicators, the indicator was conditioning actual output of the field, rather than resetting the page count. Page fields are always printed, and the output indicator's status will now be used to reset the page count. If the indicator is on, the PAGE field is reset.

Spar #3000758. Result indicators were not being turned on with the MVR calculation opcode. This has been fixed.

The maximum PRINTER file length has been increased from 132 to 198 characters.

An overflow indicator is set on when an output line writes on or past the overflow line, or spaces or skips beyond the overflow line. In previous revisions of VRPG, the overflow indicator would incorrectly turn on if the destination of the 'space after' or 'skip after' was on the overflow line.

Overflow indicators can be set on or set off when used as resulting indicators in a calculation statement. Setting on an overflow indicator is treated as though the overflow line was reached. If the overflow indicator was on before the calculation, then it is treated as though the overflow line was sensed again. This last feature was not in the previous revisions of VRPG.

Overflow processing for printer files that did not specify an overflow indicator used to skip to line 6 of a new page as soon as the overflow line was reached. This page eject should have occurred after total output, at the time of overflow output. In Rev 19.3 VRPG, all printer files that do not specify an overflow indicator are assigned an unused overflow indicator by VRPG. This indicator is used to control the skip to line 6 during overflow output, after the overflow line was reached. If more than eight printer files are used, some of the files will not be processed with overflow handling.

If more than one printer file specifies the same overflow indicator, a severity 2 error message appears. In previous revisions of VRPG, the recovery was to process both files with the same overflow indicator. At Rev 19.3, an unused overflow indicator will be assigned to the second file.

In addition to specific polers and spars fixed in this revision, a

number of the compile-time error messages for input and output statements will now specify the exact field which is causing the error. In the past, because the error message referred to the beginning line of the input or output record, it was difficult with records containing many fields to know exactly which field was creating the error.

Internal Bug Fixes

Files are now closed when terminating runtime errors occur. If a Midas file had a record locked for update, this record will be unlocked during the closing process.

An improvement has been made in output to PRINTER files. The amount of time to perform a write to a PRINTER file has been reduced.

Outstanding Problems

A data structure subfield is not accessible for evaluation or modification during a DBG session. The full data structure is accessible and thus offers the view of the subfields.

There are some outstanding problems listed on the on-line POLER*s data base.

Subject: PRIMENET

Release: Revision 19.7

Date: November 29, 1983

1 New Functionality

Half Duplex is not supported in this rev.

X.25-1980 support has been added. Fast Select facility is now available. LAPB support has been added. Level 3 timers, including call request, reset request, & clear request timeouts have been implemented. This should alleviate problems (circuit hangs) due to lost packets. Restarts are now sent based on the restart timer. Diagnostic Packets are logged in the network event log file.

Five new commands, CONFIG_NET, START_NET, STOP_NET, MONITOR_RING and FIND_RING_BREAK have been added for rev 19.3 PRIMENET.

Network start-up has been separated from system cold start. The CONFIG directive "NET ON" will no longer enable PRIMENET. To do so, you must issue the new START_NET command from the system console.

The start_net invocation may be added to your C_PRMO file. Be sure it is added BEFORE any remote disks are added. It can be added at the end of C_PRMO, after the date/time are set, so that local users can use the system while the network is initializing.

Here's how to use the new start_net command:

START_NET {configuration_filename} -NODE node_name {-Tracing_Node} {-HELP}

If no filename is specified, the file PRIMENET*>PRIMENET.CONFIG is used.

"-NODE node_name" is required on the command line because of the new global network configuration files. You should specify the local node name (the name of the node you are using).

AN EXAMPLE: if you wish to bring up node ABC on the network:

i) Use CONFIG_NET to build a network configuration file, and call it PRIMENET.CONFIG in PRIMENET*.

ii) Put the following statement in your C_PRMO file
START_NET PRIMENET*>PRIMENET.CONFIG -NODE ABC

It is safe to include both the "NET ON" directive in your CONFIG file and the START_NET command in your C_PRMO file. This will enable you to run different revs of PRIMOS without having to modify these files.

Besides starting NETMAN, the START_NET command attempts to start the ROUTE-THROUGH Server (RT_SERVER). If unsuccessful for some reason, e.g. no phantom slots available, it prints an error message on the console. The administrator can then start RT_SERVER manually by typing PH PRIMENET*>RT.COMI on the console.

The STOP_NET command is used to disable use of the network on the local node. This command effectively just logs out the network server (NETMAN), cleaning up network databases as this occurs.
Usage: STOP_NET {-HELP}

The STOP_NET command also logs out the RT_SERVER.

The Monitor_Ring command will display the statistics for the ring usage from the point of view of this node.

The Find_Ring_Break command will attempt to determine if there is a break in the ring, and its location.

2 User Visible Bug Fixes

Support for FAM-T has been removed.

The maximum number of remote login/remote log-thru users has been increased from 32 to 63.

A new diagnostic code, CD$ASV, has been added to the X$KEYS insert files. NETLINK has been updated to understand this new clearing diagnostic.

New diagnostic codes CD$RTE, CD$LOP, CD$TMO, CD$MEM and CD$RTD have been added to the X$KEYS insert files. Eventually they'll be understood as Route-Through error codes.

SPAR#2001468 : Remote login users may no longer pass off their remote login circuit.

SPAR#2005017 : The acl SLAVE*:ALL is no longer needed for PRIMENET*.

SPAR#3100148 : New format TYMNET addresses should no longer cause problems with call connections.

SPAR#3000070 : The CHAP ALL command will no longer lower NETMAN's priority.

LCN 0 is no longer used.

Call collisions are now handled correctly, with only one (not both) of the calls being cleared.

3 Internal Bug Fixes

Netlink now does the "ESCAPE FROM DATA TRANSFER" and "ACTION ON BREAK" parameter functions correctly - it is now possible for a user to disable break transmission in NETLINK, as well as disabling the escape character (usually atsign).

Impure code in VNETLB was cleaned up.

Removed obsolete CFI support: x$aofi & x$ugfi routines were removed from PRIMOS, and their dynts were removed from VNETLB.

4 Outstanding Problems

CONFIG_NET must be run on rev 19.2.2 (or later) PRIMOS, since it utilizes the new SWRAQ gate.

CONFIG_NET may encounter an internal error & abort the configuration session if you give an answer which it does not expect. If you are building a large configuration, you should SAVE periodically.

Subject: Z80MA

Release: 1.02

Z80MA is an internal tool used to assemble Z80 microprocessor code.

The changes for this release are:
1. Mic.pma was modified to allow up to 12 spaces between instruction and first operand.